xCAT3 Documentation

Release 2.12

IBM Corporation

Mar 09, 2017

Contents

xCAT stands for **Ex**treme **C**loud/**C**luster **A**dministration **T**oolkit.

xCAT offers complete management of clouds, clusters, HPC, grids, datacenters, renderfarms, online gaming infrastructure, and whatever tomorrows next buzzword may be.

xCAT enables the administrator to:

1. Discover the hardware servers

2. Execute remote system management

3. Provision operating systems on physical or virtual machines

4. Provision machines in Diskful (stateful) and Diskless (stateless)

5. Install and configure user applications

6. Parallel system management

7. Integrate xCAT in Cloud

You've reached xCAT documentation site, The main page product page is http://xcat.org

xCAT is an open source project hosted on GitHub. Go to GitHub to view the source, open issues, ask questions, and particpate in the project.

Enjoy!

Table of Contents

Admin Guide

The admin guide is intended to help with learning how to manage a cluster using xCAT with the following major sections:

- **Basic Concepts** Introduces some of the basic concepts in xCAT.

- **Manage Cluster** Describes managing clusters under xCAT. The management procedures are organized based on the hardware type since management may vary depending on the hardware architecture.

- **Reference** xCAT reference sections.

References

xCAT Man Pages

These man pages are auto generated from .pod files to .rst files using the `create_man_pages.py` script under xcat-core

man1

addkit.1

NAME

addkit - Install a kit on the xCAT management node

SYNOPSIS

addkit [**-?** | **-h** | **--help**] [**-v** | **--version**]

addkit [**-i** | **--inspection**] *kitlist*

addkit [**-V** | **--verbose**] [**-p** | **--path** *path*] *kitlist*

DESCRIPTION

The **addkit** command install a kit on the xCAT management node from a kit tarfile or directory, creating xCAT database definitions for kit, kitrepo, kitcomponent.

Note: The xCAT support for Kits is only available for Linux operating systems.

OPTIONS

-h|--help

Display usage message.

-V|--verbose

Verbose mode.

-v|--version

Command version.

-i|--inspection

Show the summary of the given kits

-p|--path <path>

The destination directory to which the contents of the kit tarfiles and/or kit deploy dirs will be copied. When this option is not specified, the default destination directory will be formed from the installdir site attribute with ./kits subdirectory.

kitlist

a comma delimited list of kit_tarball_files and kit_deploy_dirs that are to be added to the xCAT cluster. Each entry can be an absolute or relative path. For kit_tarball_files, these must be valid kits tarfiles added. For kit_deploy_dirs, these must be fully populated directory structures that are identical to the contents of an expanded kit_tarball_file.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To add two kits from tarball files.

addkit kit-test1.tar.bz2,kit-test2.tar.bz2

Output is similar to:

Kit /kit/kit-test1.tar.bz2,/kit/kit-test2.tar.bz2 was successfully added.

2. To add two kits from directories.

addkit kit-test1,kit-test2

Output is similar to:

Kit /kit/kit-test1,/kit/kit-test2 was successfully added.

3. To add a kit from tarball file to /install/test directory.

addkit -p /install/test kit-test1.tar.bz2

Output is similar to:

Kit /kit/kit-test1.tar.bz2 was successfully added.

4. To read the general infomration of the kit, without adding the kits to xCAT DB

addkit -i kit-test1.tar.bz2

Output is similar to:

kitname=xlc-12.1.0.0-Linux description=XLC12 for Linux version=12.1.0.0 ostype=Linux

SEE ALSO

lskit(1)|lskit.1, rmkit(1)|rmkit.1, addkitcomp(1)|addkitcomp.1, rmkitcomp(1)|rmkitcomp.1, chkkit-comp(1)|chkkitcomp.1

addkitcomp.1

NAME

addkitcomp - Assign Kit components to an xCAT osimage.

SYNOPSIS

addkitcomp [**-?** | **-h** | **--help**] [**-v** | **--version**]

addkitcomp [**-V** | **--verbose**] [**-a** | **--adddeps**] [**-f** | **--force**] [**-n** | **--noupgrade**] [**--noscripts**] **-i** *osimage kitcomp-name_list*

DESCRIPTION

The **addkitcomp** command will assign kit components to an xCAT osimage. The kit component meta rpm, package rpm and deploy parameters will be added to osimage's otherpkg.pkglist and postbootscripts will be added to osimages's postbootscripts attribute.

Note: The xCAT support for Kits is only available for Linux operating systems.

OPTIONS

-a|--adddeps

> Assign kitcomponent dependencies to the osimage.

-h|--help

> Display usage message.

-V|--verbose

> Verbose mode.

-v|--version

> Command version.

-f|--force

> Add kit component to osimage even if there is a mismatch in OS, version, arch, serverrole, or kitcompdeps

-i *osimage*

> The osimage name that the kit component is assigning to.

-n|--noupgrade

> Allow multiple versions of kitcomponent to be installed into the osimage, instead of kitcomponent upgrade

--noscripts

> Do not add kitcomponent's postbootscripts to osimage

kitcompname_list

> A comma-delimited list of valid full kit component names or kit component basenames that are to be added to the osimage.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To assign a kit component to osimage

addkitcomp -i rhels6.2-ppc64-netboot-compute comp-test1-1.0-1-rhels-6.2-ppc64

Output is similar to:

Assigning kit component comp-test1-1.0-1-rhels-6.2-ppc64 to osimage rhels6.2-ppc64-netboot-compute Kit components comp-test1-1.0-1-rhels-6.2-ppc64 were added to osimage rhels6.2-ppc64-netboot-compute successfully

2. To assign a kit component to osimage with its dependency.

addkitcomp -a -i rhels6.2-ppc64-netboot-compute comp-test2-1.0-1-rhels-6.2-ppc64

Output is similar to:

Assigning kit component comp-test1-1.0-1-rhels-6.0-ppc64 to osimage rhels6.2-ppc64-netboot-compute Assigning kit component comp-test2-1.0-1-rhels-6.2-ppc64 to osimage rhels6.2-ppc64-netboot-compute Kit components comp-test1-1.0-1-rhels-6.0-ppc64,comp-test2-1.0-1-rhels-6.2-ppc64 were added to osimage rhels6.2-ppc64-netboot-compute successfully

3. To assign a kit component to osimage with incompatable osarch, osversion or ostype.

addkitcomp -f -i rhels6.2-ppc64-netboot-compute comp-test1-1.0-1-rhels-6.2-ppc64

Output is similar to:

Assigning kit component comp-test1-1.0-1-rhels-6.2-ppc64 to osimage rhels6.2-ppc64-netboot-compute Kit components comp-test1-1.0-1-rhels-6.2-ppc64 were added to osimage rhels6.2-ppc64-netboot-compute successfully

4. To assign a new version of kit component to osimage without upgrade.

addkitcomp -n -i rhels6.2-ppc64-netboot-compute comp-test2-1.0-1-rhels-6.2-ppc64

Output is similar to:

Assigning kit component comp-test1-1.0-1-rhels-6.0-ppc64 to osimage rhels6.2-ppc64-netboot-compute Assigning kit component comp-test2-1.0-1-rhels-6.2-ppc64 to osimage rhels6.2-ppc64-netboot-compute Kit components comp-test2-1.0-1-rhels-6.2-ppc64 were added to osimage rhels6.2-ppc64-netboot-compute successfully

The result will be: lsdef -t osimage rhels6.2-ppc64-netboot-compute -i kitcomponents Object name: rhels6.2-ppc64-netboot-compute kitcomponents=comp-test2-1.0-0-rhels-6.2-ppc64,comp-test2-1.0-1-rhels-6.2-ppc64

SEE ALSO

lskit(1)|lskit.1, addkit(1)|addkit.1, rmkit(1)|rmkit.1, rmkitcomp(1)|rmkitcomp.1, chkkitcomp(1)|chkkitcomp.1

buildkit.1

NAME

buildkit - Used to build a software product Kit which may be used to install software in an xCAT cluster.

SYNOPSIS

buildkit [**-?** | **-h** | **--help**] [**-v** | **--version**]

To build a new Kit

buildkit [**-V** | **--verbose**] *subcommand* [*kit_name*] [*repo_name*|**all**] [**-l** | **--kitloc** *kit_location*]

To add packages to an existing Kit.

buildkit [**-V** | **--verbose**] *addpkgs kit_tarfile* [**-p** | **--pkgdir** *package_directory_list*] [**-k** | **--kitversion** *version*] [**-r** | **--kitrelease** *release*] [**-l** | **--kitloc** *kit_location*]

DESCRIPTION

The **buildkit** command provides a collection of utilities that may be used to package a software product as a Kit tarfile that can be used to install software on the nodes of an xCAT cluster. A Kit contains the product software packages, configuration and control information, and install and customization scripts.

Note: The xCAT support for Kits is only available for Linux operating systems.

You will need to run the **buildkit** command several times with different subcommands to step through the process of building a kit:

By default the **buildkit** subcommands will operate in the current working directory, (ie. look for files, create directories etc.). You could specify a different location by using the "**-l | --kitloc** *kit_location*" option.

The *kit_location* is the full path name of the directory that contains the kit files. You would use the same location value for all the buildkit subcommands.

For example, to create a new kit named "prodkit" in the directory /home/mykits/ *either* run:

1. If no location is provided then the command will create a subdirectory called "prodkit" in the current directory "/home/mykits" and the new kit files will be created there.

    cd /home/mykits

    buildkit create prodkit

    or

2. If a location is provided then the Kit files will be created there. Note that the Kit name does not necessarily have to be the directory name where the kit files are located.

    buidkit create prodkit -l /home/mykits/prodkit

In both cases the /home/mykits/prodkit directory is created and the inital files for the kit are created in that directory.

The following example illustrates the basic process for building a new Kit. In this example we are building a Kit named "mytstkit".

1. Change to the directory where you wish to create the Kit.

2. Create a template directory for your kit:

    buildkit create mytstkit

3. Change directory to the new "mytstkit" subdirectory that was just created.

    cd mytstkit

4. Edit the buildkit configuration file for your kit:

    vi buildkit.conf

    (See xCAT Kit documentation for details.)

5. Create all required files, scripts, plugins, and packages for your kit.

6. Validate your kit build configuration and fix any errors that are reported:

    buildkit chkconfig

7. List the repos defined in your buildkit configuration file:

    buildkit listrepo

8. For each repo name listed, build the repository. Note that if you need to build repositories for OS distributions, versions, or architectures that do not match the current system, you may need to copy your kit template directory to an appropriate server to build that repository, and then copy the results back to your main build server. For example, to build a repo named "rhels6.3" you would run the following command.

    buildkit buildrepo rhels6.3

    or, you can build all of the repos at one time if there are no OS or architecture dependencies for kitcomponent package builds or kitpackage builds:

    buildkit buildrepo all

9. Build the kit tar file:

> **buildkit buildtar**

OPTIONS

-h |--help

> Display usage message.

-k|--kitversion *version*

> Product version.

-l|--kitloc *kit_location*

> The directory location of the Kit files.

-p|--pkgdir *package_directory_list*

> A comma-separated list of directory locations for product RPMs.

-r|--kitrelease *release*

> Product release.

-V |--verbose

> Verbose mode.

-v|--version

> Command version.

SUB-COMMANDS

create *kit_basename*

> Creates a new kit build directory structure for kit *kit_basename* using the location specified on the command line or the current directory. The sample kit files from /opt/xcat/share/xcat/kits/kit_template are copied over, and the buildkit.conf file is modified for the specified *kit_basename*.

chkconfig

> Reads the buildkit.conf file, verifies that the file syntax is correct and that all specified files exist.

listrepo

> Reads the buildkit.conf file, lists all Kit package repositories listed in the file, and reports the build status for each repository.

buildrepo {*repo_name* | **all**}

> Reads the buildkit.conf file, and builds the specified Kit package repository. The built packages are placed in the directory <kit_location>/build/kit_repodir/*repo_name*. If **all** is specified, all kit repositories are built.

cleanrepo {*repo_name* | **all**}

> Reads the buildkit.conf file, and deletes all the package files and package meta data files from the <kit_location>/build/kit_repodir/*repo_name* directory. If **all** is specified, all kit repository files are deleted.

buildtar

Reads the buildkit.conf file, validates that all kit repositories have been built, and builds the Kit tar file <kit_location>/*kitname*.tar.bz2.

cleantar

Reads the <kit_location>/buildkit.conf file and *deletes* the following:

- Kit tar files matching <kit_location>/*kit_name*\.tar.bz2*.
- <kit_location>/build/*kit_name*
- <kit_location>/rpmbuild
- <kit_location>/tmp
- <kit_location>/debbuild

Caution: Make sure you back up any tar files you would like to keep before running this subcommand.

cleanall

Equivalent to running **buildkit cleanrepo all** and **buildkit cleantar**.

addpkgs

kit_tarfile {**-p** | **--pkgdir** *package_directory_list*} [**-k** | **--kitversion** *version*] [**-r** | **--kitrelease** *release*]

Add product package rpms to a previously built kit tar file. This is used for partial product kits that are built and shipped separately from the product packages, and are identified with a *kit_tarfile* name of *kitname*.**NEED_PRODUCT_PKGS.tar.bz2**. Optionally, change the kit release and version values when building the new kit tarfile. If kitcomponent version and/or release values are defaulted to the kit values, those will also be changed and new kitcomponent rpms will be built. If kit or kitcomponent scripts, plugins, or other files specify name, release, or version substitution strings, these will all be replaced with the new values when built into the new complete kit tarfile *kit_location*/*new_kitname*.**tar.bz2**.

RETURN VALUE

<B>0

The command completed successfully.

<B>1

An error has occurred.

EXAMPLES

1. To create the sample kit shipped with the xCAT-buildkit rpm on a RHELS 6.3 server and naming it **mykit**, run the following commands:

   cd /home/myuserid/kits

   buildkit create mykit

   cd mykit

   vi buildkit.conf

   buildkit chkconfig

   buildkit listrepo

   buildkit buildrepo all ****

buildkit buildtar

2. To clean up a kit repository directory after build failures on a RHELS 6.3 server to prepare for a new kit repository build, run:

   buildkit cleanrepo rhels6.3

3. To clean up all kit build files, including a previously built kit tar file, run

   buildkit cleanall

4. To create a kit named "tstkit" located in /home/foobar/tstkit instead of the current working directory.

   buildkit create tstkit -l /home/foobar/tstkit

FILES

/opt/xcat/bin/buildkit

/opt/xcat/share/xcat/kits/kit_template

/opt/xcat/share/xcat/kits/kitcomponent.spec.template

<kit location>/buildkit.conf

<kit location>/build/*kitname*/kit.conf

<kit location>/*kitname*.tar.bz2

SEE ALSO

addkit(1), lskit(1), rmkit(1), addkitcomp(1), rmkitcomp(1), chkkitcomp(1)

cfgve.1

NAME

cfgve - Configure the elements for a virtual environment.

SYNOPSIS

cfgve -t dc **-m** manager **-o** object [**-c -k** nfs|localfs | **-r**]

cfgve -t cl **-m** manager **-o** object [**-c -p** cpu type| **-r -f**]

cfgve -t sd **-m** manager **-o** object [**-c** | **-g** | **-s**| **-a** | **-b** | **-r -f**]

cfgve -t nw **-m** manager **-o** object [**-c -d** data center **-n**vlan ID | **-a -l** cluster | **-b** | **-r**]

cfgve -t tpl **-m** manager **-o** object [**-r**]

DESCRIPTION

The **cfgve** command can be used to configure a virtual environment for 'Storage Domain', 'Network' and 'Template' objects.

The mandatory parameter **-m manager** is used to specify the address of the manager of virtual environment. xCAT needs it to access the RHEV manager.

The mandatory parameter **-t type** is used to specify the type of the target object.

Basically, **cfgve** command supports five types of object: **dc**, **cl**, **sd**, **nw** and **tpl**.

dc - The **create** and **remove** operations are supported.

cl - The **create** and **remove** operations are supported.

sd - The **create**, **attach**, **detach**, **activate**, **deactivate** and **remove** operations are supported.

nw - The **create**, **attach**, **detach** and **remove** operations are supported.

tpl - The **remove** operation is supported.

The mandatory parameter **-o object** is used to specify which object to configure.

OPTIONS

-a To attach the target object.

-b To detach the target object.

-c To create the target object.

> For creating of **Storage Domain**, the target storage domain will be created first, then attached to data center and activated.

> The parameters that used to create the storage domain are gotten from 'virtsd' table. The detail parameters in the virtsd table:

> **virtsd.node** - The name of the storage domain.

> **virtsd.sdtype** - The type of storage domain. Valid value: data, iso, export. Default value is 'data'.

> **virtsd.stype** - The storage type. "nfs" or "localfs".

> **virtsd.location** - The location of the storage. **nfs**: Format: [nfsserver:nfspath]. The NFS export directory must be configured for read write access and must be owned by vdsm:kvm. **localfs**: "/data/images/rhev" is set by default.

> **virtsd.host** - A host must be specified for a storage doamin as SPM (Storage Pool Manager) when initialize the storage domain. The role of SPM may be migrated to other host by rhev-m during the running of the datacenter (For example, when the current SPM encountered issue or going to maintenance status.

> **virtsd.datacenter** - The storage will be attached to. 'Default' data center is the default value.

-d *data center*

> The name of data center.

> Specify the 'Data Center' that will be used for the object to be attached to. It is used by <nw> type.

-f It can be used with **-r** to remove the target object by force.

> For removing of **Storage Domain**, if **-f** is specified, the storage domain will be deactivated and detached from data center before the removing.

-g To activate the target object.

-h Display usage message.

-k *storage type*

> To specify the type of the storage type when creating the data center.

> Supported type: nfs; localfs.

-l *cluster*

> Specify the cluster for the network to attach to.

-m *manager*

> Specify the manager of the virtual environment.

> For RHEV, the FQDN (Fully Qualified Domain Name) of the rhev manager have to be specified.

-n *vlan ID*

> To specify the vlan number when creating a network.

-o *object*

> The name of the target object.

-p *cpu type*

> To specify the cpu type when creating the cluster. **Intel Penryn Family** is default type.

> Supported type: **Intel Conroe Family**, **Intel Penryn Family**, **Intel Nehalem Family**, **Intel Westmere Family**, **AMD Opteron G1**, **AMD Opteron G2**, **AMD Opteron G3**

-r To remove the target object.

> For removing of **Storage Domain**, the storage space will be formatted after removing.

-s To deactivate the target object.

-t *type*

> Specify the **type** of the target object.

> **Supported types: dc** - Data Center **cl** - Cluster **sd** - Storage Domain **nw** - Network **tpl** - Template

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To create the Storage Domain 'sd1', enter:

```
cfgve -t sd -m <FQDN of rhev manager> -o sd1 -c
```

2. To deactivate the Storage Domain 'sd1' from data center, enter:

```
cfgve -t sd -m <FQDN of rhev manager> -o sd1 -s
```

3. To remove the Storage Domain 'sd1', enter:

```
cfgve -t sd -m <FQDN of rhev manager> -o sd1 -r
```

4. To create the network 'nw1', enter:

```
cfgve -t nw -m <FQDN of rhev manager> -o nw1 -c
```

5. To remove the template 'tpl01', enter:

```
cfgve -t tpl -m <FQDN of rhev manager> -o tpl01 -r
```

FILES

/opt/xcat/bin/cfgve

SEE ALSO

lsve(1)|lsve.1

cfm2xcat.1

NAME

cfm2xcat - Migrates the CFM setup in CSM to the xdcp rsync setup in xCAT.

SYNOPSIS

cfm2xcat [**-i** *path of the CFM distribution files generated *] [*-o** **path of the xdcp rsync files generated from the CFM distribution files **]

cfm2xcat [**-h**]

DESCRIPTION

Copy the cfm2xcat command to the CSM Management Server. Run the command, indicating where you want your files saved with the -i and -o flags. They can be in the same directory. The cfm2xcat command will run cfmupdatenode -a, saving the generated CFM distribution files in the directory indicates with (-i). From those distribution files, it will generate xdcp rsync input files (-F option on xdcp) in the directory indicated by (-o). Check the rsync files generated. There will be a file generated (rsyncfiles) from the input -o option on the command, and the same file with a (.nr) extension generated for each different noderange that will used to sync files based on your CFM setup in CSM. The rsyncfiles will contain the rsync file list. The rsyncfiles.nr will contain the noderange. If multiple noderanges then the file name (rsyncfiles) will be appended with a number.

OPTIONS

-h Display usage message.

-i Path of the CFM distribution files generated from the cfmupdatenode -a command.

-o Path of the xdcp rsync input file generated from the CFM distribution files.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To build xCAT rsync files to use with xdcp -F , enter on the CSM Management Server, make sure the path exists:

cfm2xcat -i /tmp/cfm/cfmdistfiles -o /tmp/cfm/rsyncfiles

2. To use the file on the xCAT Management Node copy to /tmp/cfm on the xCAT MN:

xdcp ^/tmp/cfm/rsyncfiles.nr -F /tmp/cfm/rsyncfilesxdcp ^/tmp/cfm/rsyncfiles.nr1 -F /tmp/cfm/rsyncfiles1xdcp ^/tmp/cfm/rsyncfiles.nr2 -F /tmp/cfm/rsyncfiles2

FILES

/opt/xcat/share/xcat/tools/cfm2xcat

chdef.1

NAME

chdef - Change xCAT data object definitions.

SYNOPSIS

chdef [**-h | --help**] [**-t** *object-types*]

chdef [**-t** *object-types*] [**-o** *object-names*] [**-n** *new-name*] [*node*]

chdef [**-V | --verbose**] [**-t** *object-types*] [**-o** *object-names*] [**-d | --dynamic**] [**-p | --plus**] [**-m | --minus**] [**-z | --stanza**] [[**-w** *attr==val*] [**-w** *attr=~val*] ...] [*noderange*] [*attr=val* [*attr=val...*]]

> [**-u** [*provmethod=<install|netboot|statelite>*] [*profile=<xxx>*] [*osvers=value*] [*osarch=value*]]

DESCRIPTION

This command is used to change xCAT object definitions which are stored in the xCAT database. The default is to replace any existing attribute value with the one specified on the command line. The command will also create a new definition if one doesn't exist.

This command also can be used to change the xCAT object name to a new name. Note: the site,monitoring types can NOT be supported.

OPTIONS

attr=val [attr=val ...]

> Specifies one or more "attribute equals value" pairs, separated by spaces. Attr=val pairs must be specified last on the command line. Use the help option to get a list of valid attributes for each object type.

-d|--dynamic

> Use the dynamic option to change dynamic node groups definition. This option must be used with -w option.

-h|--help

> Display usage message.

-m|--minus

> If the value of the attribute is a list then this option may be used to remove one or more items from the list.

-n *new-name*

> Change the current object name to the new-name which is specified by the -n option. Objects of type site and monitoring cannot be renamed with the -n option. Note: For the **-n** option, only one node can be specified. For some special nodes such as fsp, bpa, frame, cec etc., their name is referenced in their own hcp attribute, or the hcp attribute of other nodes. If you use **-n** option, you must manually change all hcp attributes that refer to this name.

noderange

> A set of comma delimited node names and/or group names. (must be the first parameter) See the "noderange" man page for details on supported formats.

-o *object-names*

> A set of comma delimited object names.

-p|--plus

> This option will add the specified values to the existing value of the attribute. It will create a comma-separated list of values.

-t *object-types*

> A set of comma delimited object types. Use the help option to get a list of valid object types.

-V|--verbose

> Verbose mode.

-w *attr==val* **-w** *attr=~val ...*

> Use one or multiple -w flags to specify the selection string that can be used to select objects. The operators ==, !=, =~ and !~ are available. Use the help option to get a list of valid attributes for each object type.

> **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.

> Note: the operator !~ will be parsed by shell, if you want to use !~ in the selection string, use single quote instead. For example:-w 'mgt!~ipmi'.

-z|--stanza

Indicates that the file being piped to the command is in stanza format. See the xcatstanzafile man page for details on using xCAT stanza files.

-u

Fill in the attributes such as template file, pkglist file and otherpkglist file of osimage object based on the specified parameters. It will search "/install/custom/" directory first, and then "/opt/xcat/share/".

Note: this option only works for objtype **osimage**.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To change a site definition.

```
chdef -t site -o clustersite installdir=/xcatinstall
```

2. To change a basic node definition.

```
chdef -t node -o node01 groups="all,aix"
```

(The group definitions are also created if they don't already exist.)

3. To add another group to the "groups" attribute in the previous example.

```
chdef -p -t node -o node01 groups="compute"
```

4. To remove the "all" group from the "groups" attribute in the previous example.

```
chdef -m -t node -o node01 groups="all"
```

5. To replace the current "groups" attribute value of "node01".

```
chdef -t node -o node01 groups="linux"
```

6. To add "node01" to the "members" attribute of a group definition called "LinuxNodes".

```
chdef -p -t group -o LinuxNodes members="node01"
```

7. To update a set of definitions based on information contained in the stanza file mystanzafile.

```
cat mystanzafile | chdef -z
```

8. To update a dynamic node group definition to add the cons=hmc wherevals pair.

```
chdef -t group -o dyngrp -d -p -w cons==hmc
```

9. To change the node object name from node1 to node2.

```
chdef -t node -o node1 -n node2
```

10. To change the node hwtype, this command will change the value of ppc.nodetype.

```
chdef -t node -o node1 hwtype=lpar
```

11. To change the policy table for policy number 7.0 for admin1

```
chdef -t policy -o 7.0 name=admin1 rule=allow
```

12. To change the node nic attributes

```
chdef -t node -o cn1 nicips.eth0="1.1.1.1|1.2.1.1" nicnetworks.eth0="net1|net2"␣
↪nictypes.eth0="Ethernet"
```

13. To update an osimage definition.

```
chdef redhat6img -u provmethod=install
```

FILES

$XCATROOT/bin/chdef

(The XCATROOT environment variable is set when xCAT is installed. The default value is "/opt/xcat".)

NOTES

This command is part of the xCAT software product.

SEE ALSO

mkdef(1)|mkdef.1, lsdef(1)|lsdef.1, rmdef(1)|rmdef.1, xcatstanzafile(5)|xcatstanzafile.5

chhypervisor.1

NAME

chhypervisor - Configure the virtualization hosts.

SYNOPSIS

RHEV specific :

chhypervisor *noderange* [**-a**]

chhypervisor *noderange* [**-n**]

chhypervisor *noderange* [**-p**]

chhypervisor *noderange* [**-e**]

chhypervisor *noderange* [**-d**]

zVM specific :

chhypervisor *noderange* [**--adddisk2pool** *function region volume group*]

chhypervisor *noderange* [**--addscsi** *device_number device_path option persist*]

chhypervisor *noderange* [**--addvlan** *name owner type transport*]

chhypervisor *noderange* [**--addvswitch** *name osa_dev_addr osa_exp_adapter controller connect (0, 1, or 2) memory_queue router transport vlan_id port_type update gvrp native_vlan*]

chhypervisor *noderange* [**--addzfcp2pool** *pool status wwpn lun size owner*]

chhypervisor *noderange* [**--removediskfrompool** *function region group*]

chhypervisor *noderange* [**--removescsi** *device_number persist (YES or NO)*]

chhypervisor *noderange* [**--removevlan** *name owner*]

chhypervisor *noderange* [**--removevswitch** *name*]

chhypervisor *noderange* [**--removezfcpfrompool** *pool lun wwpn*]

chhypervisor *noderange* [**--smcli** *function arguments*]

DESCRIPTION

The **chhypervisor** command can be used to configure the RHEV-h.

The rhev-h host will register to the rhev-m automatically, but admin needs to approve the host can be added to the 'cluster' with **-a** flag .

After registering, the network interfaces of host need to be added to the 'network' of RHEV. And the power management for the host should be configured so that rhev-m could make proper decision when certain host encountered error.

The **chhypervisor** command can also be used to configure the zVM host.

For each host, an entry should be added to the hypervisor table:

The columns of hypervisor table:

hypervisor.node - rhev-h host name or zVM host name (lower-case).

hypervisor.type - Must be set to 'rhevh' or 'zvm'.

hypervisor.mgr - The rhev manager (The FQDN of rhev-m server) for the host.

hypervisor.interface - The configuration for the nics. Refer to **-n**.

hypervisor.cluster - The cluster that the host will be added to. The default is 'Default' cluster if not specified.

OPTIONS

RHEV specific :

-a Approve the host that to be added to cluster.

> Before approve, the status of the host must be 'pending_approval'.

-n Configure the network interfaces for the host.

> Note: This operation only can be run when host is in 'maintenance mode'. Use **-d** to switch the host to 'maintenance' mode.

> The interfaces which configured in hypervisor.interface will be added to the network of RHEV.

The format of hypervisor.interface is multiple [network:interfacename: protocol:IP:netmask:gateway] sections separated with '|'. For example: [rhevm2:eth0:static:10.1.0.236:255.255.255.0:0.0.0.0].

network - The logic network which has been created by 'cfgve -t nw' or the default management network 'rhevm'.

interfacename - Physical network name: 'eth0','eth1'...

protocol - To identify which boot protocol to use for the interface: dhcp or static.

IP - The IP address for the interface.

netmask - The network mask for the interface.

gateway - The gateay for the interface. This field only can be set when the interface is added to 'rhevm' network.

-p Configure the power management for the host.

The power management must be configured for the rhev-h host to make the rhev-m to monitor the power status of the host, so that when certain host failed to function, rhev-m will fail over certain role like SPM to other active host.

For rack mounted server, the bmc IP and user:password need to be set for the power management (These parameters are gotten from ipmi table). rhev-m uses the ipmi protocol to get the power status of the host.

-e To activate the host.

-d To deactivate the host to maintenance mode.

-h Display usage message.

zVM specific :

--adddisk2pool *function region volume group*

Add a disk to a disk pool defined in the EXTENT CONTROL. Function type can be either: (4) Define region as full volume and add to group OR (5) Add existing region to group. If the volume already exists in the EXTENT CONTROL, use function 5. If the volume does not exist in the EXTENT CONTROL, but is attached to SYSTEM, use function 4.

--addscsi *device_number device_path option persist*

Dynamically add a SCSI disk to a running z/VM system.

--addvlan *name owner type transport*

Create a virtual network LAN.

--addvswitch *name osa_dev_addr osa_exp_adapter controller connect (0, 1, or 2) memory_queue router transport vlan_id port_type update gvrp native_vlan*

Create a virtual switch.

--addzfcp2pool *pool status wwpn lun size owner*

Add a zFCP device to a device pool defined in xCAT. The device must have been carved up in the storage controller and configured with a WWPN/LUN before it can be added to the xCAT storage pool. z/VM does not have the ability to communicate directly with the storage controller to carve up disks dynamically.

--removediskfrompool *function region group*

Remove a disk from a disk pool defined in the EXTENT CONTROL. Function type can be either: (1) Remove region, (2) Remove region from group, (3) Remove region from all groups, OR (7) Remove entire group .

--removescsi *device_number persist (YES or NO)*

Delete a real SCSI disk.

--removevlan *name owner*

Delete a virtual network LAN.

--removevswitch *name*

Delete a virtual switch.

--removezfcpfrompool *pool lun*

Remove a zFCP device from a device pool defined in xCAT.

--smcli *function arguments*

Execute a SMAPI function. A list of APIs supported can be found by using the help flag, e.g. chhypervisor pokdev61 –smcli -h. Specific arguments associated with a SMAPI function can be found by using the help flag for the function, e.g. chhypervisor pokdev61 –smcli Image_Query_DM -h. Only z/VM 6.2 and older SMAPI functions are supported at this time. Additional SMAPI functions will be added in subsequent zHCP versions.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

RHEV specific :

1. To approve the host 'host1', enter:

```
chhypervisor host1 -a
```

2. To configure the network interface for the host 'host1', enter:

```
chhypervisor host1 -n
```

3. To configure the power management for the host 'host1', enter:

```
chhypervisor host1 -p
```

4. To activate the host 'host1', enter:

```
chhypervisor host1 -e
```

5. To deactivate the host 'host1', enter:

```
chhypervisor host1 -d
```

zVM specific :

1. To add a disk to a disk pool defined in the EXTENT CONTROL, enter:

```
chhypervisor pokdev61 --adddisk2pool 4 DM1234 DM1234 POOL1
```

2. To add a zFCP device to a device pool defined in xCAT, enter:

```
chhypervisor pokdev61 --addzfcp2pool zfcp1 free 500501234567C890␣
↪4012345600000000 8G
```

3. To remove a region from a group in the EXTENT CONTROL, enter:

```
chhypervisor pokdev61 --removediskfrompool 2 DM1234 POOL1
```

4. To remove a zFCP device from a device pool defined in xCAT, enter:

```
chhypervisor pokdev61 --removezfcpfrompool zfcp1 4012345600000000␣
↪500501234567C890
```

5. To execute a SMAPI function (Image_Query_DM), enter:

```
chhypervisor pokdev61 --smcli Image_Query_DM -T LNX3
```

FILES

/opt/xcat/bin/chhypervisor

chkkitcomp.1

NAME

chkkitcomp - Check if Kit components are compatible with an xCAT osimage.

SYNOPSIS

chkkitcomp [**-?** | **-h** | **--help**] [**-v** | **--version**]

chkkitcomp [**-V** | **--verbose**] **-i** *osimage kitcompname_list*

DESCRIPTION

The **chkkitcomp** command will check if the kit components are compatible with the xCAT osimage.

This command will ignore the current osimage.kitcomponents setting, and just to check if the kitcompname list in the cmdline are compatible with the osimage by osversion/ostype/osarch/ and kit component dependencies.

Note: The xCAT support for Kits is only available for Linux operating systems.

OPTIONS

-h|--help

> Display usage message.

-V|--verbose

> Verbose mode.

-v|--version

> Command version.

-i *osimage*

> The name of the osimage is used for check.

kitcompname_list

> A comma-delimited list of valid full kit component names or kit component basenames that are to be checking to the osimage.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To check if a kit component is fitting to an osimage

chkkitcomp -i rhels6.2-ppc64-netboot-compute comp-test1-1.0-1-rhels-6.2-ppc64

Output is similar to:

Kit components comp-test1-1.0-1-rhels-6.2-ppc64 fit to osimage rhels6.2-ppc64-netboot-compute

SEE ALSO

lskit(1)|lskit.1, addkit(1)|addkit.1, rmkit(1)|rmkit.1, addkitcomp(1)|addkitcomp.1, rmkitcomp(1)|rmkitcomp.1

chkosimage.1

NAME

chkosimage - Use this xCAT command to check an xCAT osimage.

SYNOPSIS

chkosimage [-h | --help]

chkosimage [-V] [-c|--clean] osimage_name

DESCRIPTION

This command is currently supported for AIX osimages only.

Use this command to verify if the NIM lpp_source directories contain the correct software. The lpp_source directory must contain all the software that is specified in the "installp_bundle" and "otherpkgs" attributes of the osimage definition.

The command gets the name of the lpp_source resource from the xCAT osimage definition and the location of the lpp_source directory from the NIM resource definition.

It will check for installp, rpm and emgr type packages.

Note: Remember to use the prefixes, "I:", "R:", and "E:", respectively, when specifying package names in an installp_bundle file or an otherpkgs list.

In addition to checking for missing software the chkosimage command will also check to see if there are multiple matches. This could happen when you use wildcards in the software file names. For example, if you have perl-xCAT* in a bundle file it could match multiple versions of the xCAT rpm package saved in your lpp_source directory.

If this happens you must remove the unwanted versions of the rpms. If the extra rpms are not removed you will get install errors.

To help with this process you can use the "-c|–clean" option. This option will keep the rpm package with the most recent timestamp and remove the others.

The chkosimage command should always be used to verify the lpp_source content before using the osimage to install any AIX cluster nodes.

OPTIONS

-c |--clean

Remove any older versions of the rpms. Keep the version with the latest timestamp.

-h |--help

Display usage message.

osimage_name

The name of the xCAT for AIX osimage definition.

-V |--verbose

Verbose mode.

RETURN VALUE

0

The command completed successfully.

1

An error has occurred.

EXAMPLES

1

Check the XCAT osimage called "61image" to verify that the lpp_source directories contain all the software that is specified in the "installp_bundle" and "otherpkgs" attributes.

chkosimage -V 61image

2

Clean up the lpp_source directory for the osimage named "61img" by removing any older rpms with the same names but different versions.

chkosimage -c 61img

FILES

/opt/xcat/bin/chkosimage

NOTES

This command is part of the xCAT software product.

SEE ALSO

mknimimage(1)|mknimimage.1

chvlan.1

NAME

chvlan - It adds or removes nodes for the vlan.

SYNOPSIS

chvlan *vlanid* **-n | --nodes** *noderange* [**-i | --interface** *nic*]

chvlan *vlanid* **-n | --nodes** *noderange* **-d | --delete**

chvlan [**-h | --help**]

chvlan [**-v | --version**]

DESCRIPTION

The **chvlan** command adds nodes to the given vlan. If -d is specified, the nodes will be removed from the vlan.

For added security, the root guard and bpdu guard will be enabled for the ports added to this vlan. However, the guards will not be disabled if the ports are removed from the vlan using chvlan (-d) or rmvlan commands. To disable them, you need to use the switch command line interface. Please refer to the switch command line interface manual to see how to disable the root guard and bpdu guard for a port.

Parameters

vlanid is a unique vlan number.

OPTIONS

-n|--nodes The nodes or groups to be added or removed. It can be stand alone nodes or KVM guests. It takes the noderange format. Please check the man page for noderange for details.

-i|--interface (For adding only). The interface name where the vlan will be tagged on. If omitted, the xCAT management network will be assumed. For KVM, it is the interface name on the host.

-h|--help Display usage message.

-v|--version The Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To add node1, node2 and node3 to vlan 3.

```
chvlan 3 -n node1,node2,node3
```

2. To add node1, node2 and node3 to vlan 3 using eth1 interface.

```
chvlan 3 -n node1,node2,node3 -i eth1
```

3. TO remove node1, node2 and node3 from vlan 3.

```
chvlan -n node1,node2,node3 -d
```

4. To add KVM guests node1 and node2 to vlan 3

```
mkdef node1 arch=x86_64 groups=kvm,all installnic=mac primarynic=mac mgt=kvm
→netboot=pxe nfsserver=10.1.0.204 os=rhels6 profile=compute provmethod=install
→serialport=0 serialspeed=115200 vmcpus=1 vmhost=x3650n01 vmmemory=512
→vmnics=br0 vmstorage=nfs://10.1.0.203/vms

mkdef node2 arch=x86_64 groups=kvm,all installnic=mac primarynic=mac mgt=kvm
→netboot=pxe nfsserver=10.1.0.204 os=rhels6 profile=compute provmethod=install
→serialport=0 serialspeed=115200 vmcpus=1 vmhost=x3650n01 vmmemory=512
→vmnics=br0 vmstorage=nfs://10.1.0.203/vms

chvlan 3 -n node1,node2

mkvm node1,node2 -s 20G

rpower node1,node2 on

rinstall node1,node2
```

5. To remove KVM guests node1 and node2 from vlan 3

```
chvlan 3 -n node1,node2 -d

rpower node1,node2 off

rmvm node1,node2
```

FILES

/opt/xcat/bin/chvlan

SEE ALSO

mkvlan(1)|mkvlan.1, rmvlan(1)|rmvlan.1, lsvlan(1)|lsvlan.1

chvlanports.1

NAME

chvlanports - It adds or removes nodes' switch interfaces for the vlan.

SYNOPSIS

chvlanports *vlanid* **-n | --nodes** *noderange* **-i | --interface** *nic*

chvlanports *vlanid* **-n | --nodes** *noderange* **-i | --interface** *nic* **-d | --delete**

chvlanports [**-h | --help**]

chvlanports [**-v | --version**]

DESCRIPTION

The **chvlanports** command adds nodes switch interfaces to the given vlan. If -d is specified, the nodes switch interfaces will be removed from the vlan.

This command won't create/remove vlans on switches, it just add node's switch ports into exisitng vlan or remove them from existing vlan on switch. Before calling chvlanports, the nodes switch interfaces should be configured in table switch, and vlan must already existing in switches. =head1 Parameters

vlanid is a unique vlan number.

OPTIONS

-n|--nodes The nodes or groups to be added or removed. It takes the noderange format. Please check the man page for noderange for details.

-i|--interface The interface name where the vlan will be tagged on.

-h|--help Display usage message.

-v|--version The Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To add node1, node2 and node3 to vlan 3 using eth1 interface.

   ```
   chvlanports 3 -n node1,node2,node3 -i eth1
   ```

2. TO remove eth1 interface of node1, node2 and node3 from vlan 3.

   ```
   chvlanports 3 -n node1,node2,node3 -i eth1 -d
   ```

FILES

/opt/xcat/bin/chvlanports

SEE ALSO

mkvlan(1)|mkvlan.1, rmvlan(1)|rmvlan.1, lsvlan(1)|lsvlan.1, chvlan(1)|chvlan.1

chvm.1

NAME

chvm - Changes HMC-, DFM-, IVM-, and zVM-managed partition profiles or virtual machines. For Power 775, chvm could be used to change the octant configuration values for generating LPARs; change the I/O slots assignment to LPARs within the same CEC.

SYNOPSIS

chvm [**-h**| **--help**]

chvm [**-v**| **--version**]

PPC (with HMC) specific:

chvm [**-V**| **--verbose**] *noderange* [**-p** *profile*]

chvm [**-V**| **--verbose**] *noderange attr=val* [*attr=val*...]

PPC (using Direct FSP Management) specific:

chvm *noderange* -*-**p775** [**-**p** *profile*]

chvm *noderange* -*-**p775** **-i id** [-**m** *memory_interleaving*] -**r** *partition_rule*

chvm *noderange* [**lparname={* | name}**]

chvm *noderange* [**vmcpus=min/req/max**] [**vmmemory=min/req/max**] [**vmothersetting=hugepage:N,bsr:N**]
[**add_physlots=drc_index1,drc_index2...**]   [**add_vmnics=vlan1[,vlan2..]]**]
[**add_vmstorage=<N|viosnode:slotid>**]   [**--vios**]   [**del_physlots=drc_index1,drc_index2...**]
[**del_vadapter=slotid**]

VMware/KVM specific:

chvm *noderange* [-**a** *size*] [-**d** *disk*] [-**p** *disk*] [**--resize disk=**size] [**--cpus** *count*] [**--mem** *memory*]

zVM specific:

chvm *noderange* [**--add3390** *disk_pool device_address size mode read_password write_password multi_password*]

chvm *noderange* [**--add3390active** *device_address mode*]

chvm *noderange* [**--add9336** *disk_pool device_address size mode read_password write_password multi_password*]

chvm *noderange* [**--adddisk2pool** *function region volume group*]

chvm *noderange* [**--addnic** *device_address type device_count*]

chvm *noderange* [**--addpagespool** *volume_address volume_label volume_use system_config_name system_config_type parm_disk_owner parm_disk_number parm_disk_password*]

chvm *noderange* [**--addprocessor** *device_address*]

chvm *noderange* [**--addprocessoractive** *device_address type*]

chvm *noderange* [**--addvdisk** *device_address size*]

chvm *noderange* [**--addzfcp** *pool device_address loaddev size tag wwpn lun*]

chvm *noderange* [**--connectnic2guestlan** *device_address lan owner*]

chvm *noderange* [**--connectnic2vswitch** *device_address vswitch*]

chvm *noderange* [**--copydisk** *target_address source_node source_address*]

chvm *noderange* [**--dedicatedevice** *virtual_device real_device mode*]

chvm *noderange* [**--deleteipl**]

chvm *noderange* [**--disconnectnic** *device_address*]

chvm *noderange* [**--formatdisk** *device_address multi_password*]

chvm *noderange* [**--grantvswitch** *vswitch*]

chvm *noderange* [**--purgerdr**]

chvm *noderange* [**--removedisk** *device_address*]

chvm *noderange* [**--removenic** *device_address*]

chvm *noderange* [**--removeprocessor** *device_address*]

chvm *noderange* [**--removeloaddev** *wwpn lun*]

chvm *noderange* [**--removezfcp** *device_address wwpn lun*]

chvm *noderange* [**--replacevs** *directory_entry*]

chvm *noderange* [**--setipl** *ipl_target load_parms parms*]

chvm *noderange* [**--setpassword** *password*]

chvm *noderange* [**--setloaddev** *wwpn lun*]

chvm *noderange* [**--sharevolume** *volume_address share_enable*]

chvm *noderange* [**--undedicatedevice** *device_address*]

DESCRIPTION

PPC (with HMC) specific:

The chvm command modifies the partition profile for the partitions specified in noderange. A partitions current profile can be read using lsvm, modified, and piped into the chvm command, or changed with the -p flag.

This command also supports to change specific partition attributes by specifying one or more "attribute equals value" pairs in command line directly, without whole partition profile.

PPC (using Direct FSP Management) specific:

For Power 755(use option –p775 to specify):

chvm could be used to change the octant configuration values for generating LPARs. chvm is designed to set the Octant configure value to split the CPU and memory for partitions, and set Octant Memory interleaving value. The chvm will only set the pending attributes value. After chvm, the CEC needs to be rebooted manually for the pending values to be enabled. Before reboot the cec, the administrator can use chvm to change the partition plan. If the the partition needs I/O slots, the administrator should use chvm to assign the I/O slots.

chvm is also designed to assign the I/O slots to the new LPAR. Both the current IO owning lpar and the new IO owning lpar must be powered off before an IO assignment. Otherwise, if the I/O slot is belonged to an Lpar and the LPAR is power on, the command will return an error when trying to assign that slot to a different lpar.

The administrator should use lsvm to get the profile content, and then edit the content, and add the node name with ":" manually before the I/O which will be assigned to the node. And then the profile can be piped into the chvm command, or changed with the -p flag.

For normal power machine:

chvm could be used to modify the resources assigned to partitions. The admin shall specify the attributes with options *vmcpus*, *vmmemory*, *add_physlots*, *vmothersetting*, *add_vmnics* and/or *add_vmstorage*. If nothing specified, nothing will be returned.

VMware/KVM specific:

The chvm command modifes the vm specified in noderange. Calling with deregister or purge options at the same time as the resize option is not recommended.

zVM specific:

The chvm command modifes the virtual machine's configuration specified in noderange.

OPTIONS

Common:

-h

> Display usage message.

-v

> Command Version.

PPC (with HMC) specific:

-p *profile*

> Name of an existing partition profile.

attr=val

> Specifies one or more "attribute equals value" pairs, separated by spaces.

-V

> Verbose output.

PPC (using Direct FSP Management) specific:

--p775

> Specify the operation is for Power 775 machines.

-i

> Starting numeric id of the newly created partitions. For Power 775 using Direct FSP Management, the id value only could be **1**, **5**, **9**, **13**, **17**, **21**, **25** and **29**. Shall work with option **--p775**.

-m

> memory interleaving. The setting value only could be **1** or **2**. **2** means **non-interleaved** mode (also 2MC mode), the memory cannot be shared across the processors in an octant. **1** means **interleaved** mode (also 8MC mode) , the memory can be shared. The default value is **1**. Shall work with option **--p775**.

-r

> partition rule. Shall work with option **--p775**.
>
> If all the octants configuration value are same in one CEC, it will be " **-r 0-7**:*value*" .
>
> If the octants use the different configuration value in one cec, it will be "**-r 0**:*value1*,**1**:*value2*,...**7**:*value7*", or "**-r 0**:*value1*,**1-7**:*value2*" and so on.
>
> The octants configuration value for one Octant could be **1**, **2**, **3**, **4**, **5**. The meanings of the octants configuration value are as following:

```
1 -- 1 partition with all cpus and memory of the octant
2 -- 2 partitions with a 50/50 split of cpus and memory
3 -- 3 partitions with a 25/25/50 split of cpus and memory
4 -- 4 partitions with a 25/25/25/25 split of cpus and memory
5 -- 2 partitions with a 25/75 split of cpus and memory
```

-p *profile*

> Name of I/O slots assignment profile. Shall work with option **--p775**.

lparname={* | name}

> Set LPAR name for the specified lpars. If '*' specified, it means to get names from xCAT database and then set them for the specified lpars. If a string is specified, it only supports single node and the string will be set for the specified lpar. The user can use lsvm to check the lparnames for lpars.

vmcpus=value vmmemory=value add_physlots=value vmothersetting=value

> To specify the parameters that will be modified.

add_vmnics=value add_vmstorage=value [--vios]

> To create new virtual adapter for the specified node.

del_physlots=drc_index1,drc_index2...

> To delete physical slots which are specified by the *drc_index1,drc_index2...*.

del_vadapter=slotid

> To delete a virtual adapter specified by the *slotid*.

VMware/KVM specific:

-a *size*

> Add a new Hard disk with size defaulting to GB. Multiple can be added with comma separated values.

--cpus *count*

> Set the number of CPUs.

-d *disk*

> Deregister the Hard disk but leave the backing files. Multiple can be done with comma separated values. The disks are specified by SCSI id. Size defaults to GB.

--mem *memory*

> Set the memory, defaults to MB.

-p *disk*

> Purge the Hard disk. Deregisters and deletes the files. Multiple can be done with comma separated values. The disks are specified by SCSI id. Size defaults to GB.

--resize disk=*size*

> Change the size of the Hard disk. The disk can never be set to less than it's current size. Multiple disks can be resized to *size* by using comma separated values on the left side of **=**. The disks are specified by SCSI id. Size defaults to GB.

zVM specific:

--add3390 *disk_pool device_address size mode read_password write_password multi_password*

> Adds a 3390 (ECKD) disk to a virtual machine's directory entry. The device address can be automatically assigned by specifying 'auto'. The size of the disk can be specified in GB, MB, or the number of cylinders.

--add3390active *device_address mode*

> Adds a 3390 (ECKD) disk that is defined in a virtual machine's directory entry to that virtual server's active configuration.

--add9336 *disk_pool device_address size mode read_password write_password multi_password*

> Adds a 9336 (FBA) disk to a virtual machine's directory entry. The device address can be automatically assigned by specifying 'auto'. The size of the disk can be specified in GB, MB, or the number of blocks.

--adddisk2pool *function region volume group*

> Add a disk to a disk pool defined in the EXTENT CONTROL. Function type can be either: (4) Define region as full volume and add to group OR (5) Add existing region to group. The disk has to already be attached to SYSTEM.

--addnic *device_address type device_count*

> Adds a network adapter to a virtual machine's directory entry (case sensitive).

--addpagespool *volume_addr volume_label volume_use system_config_name system_config_type parm_disk_owner parm_disk_number parm_disk_password*

> Add a full volume page or spool disk to the virtual machine.

--addprocessor *device_address*

> Adds a virtual processor to a virtual machine's directory entry.

--addprocessoractive *device_address type*

> Adds a virtual processor to a virtual machine's active configuration (case sensitive).

--addvdisk *device_address size*

> Adds a v-disk to a virtual machine's directory entry.

--addzfcp *pool device_address loaddev size tag wwpn lun*

> Add a zFCP device to a device pool defined in xCAT. The device must have been carved up in the storage controller and configured with a WWPN/LUN before it can be added to the xCAT storage pool. z/VM does not have the ability to communicate directly with the storage controller to carve up disks dynamically. xCAT will find the a zFCP device in the specified pool that meets the size required, if the WWPN and LUN are not given. The device address can be automatically assigned by specifying 'auto'. The WWPN/LUN can be set as the LOADDEV in the directory entry if (1) is specified as the 'loaddev'.

--connectnic2guestlan *device_address lan owner*

> Connects a given network adapter to a GuestLAN.

--connectnic2vswitch *device_address vswitch*

> Connects a given network adapter to a VSwitch.

--copydisk *target_address source_node source_address*

> Copy a disk attached to a given virtual server.

--dedicatedevice *virtual_device real_device mode*

Adds a dedicated device to a virtual machine's directory entry.

--deleteipl

Deletes the IPL statement from the virtual machine's directory entry.

--disconnectnic *device_address*

Disconnects a given network adapter.

--formatdisk *disk_address multi_password*

Formats a disk attached to a given virtual server (only ECKD disks supported). The disk should not be linked to any other virtual server. This command is best used after add3390().

--grantvswitch *vswitch*

Grant vSwitch access for given virtual machine.

--purgerdr

Purge the reader belonging to the virtual machine

--removedisk *device_address*

Removes a minidisk from a virtual machine's directory entry.

--removenic *device_address*

Removes a network adapter from a virtual machine's directory entry.

--removeprocessor *device_address*

Removes a processor from an active virtual machine's configuration.

--removeloaddev *wwpn lun*

Removes the LOADDEV statement from a virtual machines's directory entry.

--removezfcp *device_address wwpn lun*

Removes a given SCSI/FCP device belonging to the virtual machine.

--replacevs *directory_entry*

Replaces a virtual machine's directory entry. The directory entry can be echoed into stdin or a text file.

--setipl *ipl_target load_parms parms*

Sets the IPL statement for a given virtual machine.

--setpassword *password*

Sets the password for a given virtual machine.

--setloaddev *wwpn lun*

Sets the LOADDEV statement in the virtual machine's directory entry.

--undedicatedevice *device_address*

Delete a dedicated device from a virtual machine's active configuration and directory entry.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

PPC (with HMC) specific:

1. To change the partition profile for lpar4 using the configuration data in the file /tmp/lparfile, enter:

```
cat /tmp/lparfile | chvm lpar4
```

Output is similar to:

```
lpar4: Success
```

2. To change the partition profile for lpar4 to the existing profile 'prof1', enter:

```
chvm lpar4 -p prof1
```

Output is similar to:

```
lpar4: Success
```

3. To change partition attributes for lpar4 by specifying attribute value pairs in command line, enter:

```
chvm lpar4 max_mem=4096
```

Output is similar to:

```
lpar4: Success
```

PPC (using Direct FSP Management) specific:

1. For Power 775, to create a new partition lpar1 on the first octant of the cec cec01, lpar1 will use all the cpu and memory of the octant 0, enter:

```
mkdef -t node -o lpar1 mgt=fsp groups=all parent=cec01    nodetype=lpar    hcp=cec01
```

then:

```
chvm lpar1 --p775 -i 1 -m 1 -r 0:1
```

Output is similar to:

```
lpar1: Success
cec01: Please reboot the CEC cec1 firstly, and then use chvm to assign the I/O slots
→to the LPARs
```

2. For Power 775, to create new partitions lpar1-lpar8 on the whole cec cec01, each LPAR will use all the cpu and memory of each octant, enter:

```
mkdef -t node -o lpar1-lpar8 nodetype=lpar  mgt=fsp groups=all parent=cec01  hcp=cec01
```

then:

```
chvm lpar1-lpar8 --p775 -i 1 -m 1 -r 0-7:1
```

Output is similar to:

```
lpar1: Success
lpar2: Success
lpar3: Success
lpar4: Success
lpar5: Success
lpar6: Success
lpar7: Success
lpar8: Success
cec01: Please reboot the CEC cec1 firstly, and then use chvm to assign the I/O slots␣
↪to the LPARs
```

3. For Power 775 cec1, to create new partitions lpar1-lpar9, the lpar1 will use 25% CPU and 25% memory of the first octant, and lpar2 will use the left CPU and memory of the first octant. lpar3-lpar9 will use all the cpu and memory of each octant, enter:

```
mkdef -t node -o lpar1-lpar9 mgt=fsp groups=all parent=cec1    nodetype=lpar    hcp=cec1
```

then:

```
chvm lpar1-lpar9 --p775 -i 1 -m 1  -r 0:5,1-7:1
```

Output is similar to:

```
lpar1: Success
lpar2: Success
lpar3: Success
lpar4: Success
lpar5: Success
lpar6: Success
lpar7: Success
lpar8: Success
lpar9: Success
cec1: Please reboot the CEC cec1 firstly, and then use chvm to assign the I/O slots␣
↪to the LPARs
```

4.To change the I/O slot profile for lpar4 using the configuration data in the file /tmp/lparfile, the I/O slots information is similar to:

```
4: 514/U78A9.001.0123456-P1-C17/0x21010202/2/1
4: 513/U78A9.001.0123456-P1-C15/0x21010201/2/1
4: 512/U78A9.001.0123456-P1-C16/0x21010200/2/1
```

then run the command:

```
cat /tmp/lparfile | chvm lpar4 --p775
```

5. To change the I/O slot profile for lpar1-lpar8 using the configuration data in the file /tmp/lparfile. Users can use the output of lsvm.and remove the cec information, and modify the lpar id before each I/O, and run the command as following:

```
chvm lpar1-lpar8 --p775 -p /tmp/lparfile
```

6. To change the LPAR name, enter:

```
chvm lpar1 lparname=test_lpar01
```

Output is similar to:

```
lpar1: Success
```

7. For Normal Power machine, to modify the resource assigned to a partition:

Before modify, the resource assigned to node 'lpar1' can be shown with: lsvm lpar1

The output is similar to:

```
lpar1: Lpar Processor Info:
Curr Processor Min: 1.
Curr Processor Req: 4.
Curr Processor Max: 16.
lpar1: Lpar Memory Info:
Curr Memory Min: 1.00 GB(4 regions).
Curr Memory Req: 4.00 GB(16 regions).
Curr Memory Max: 32.00 GB(128 regions).
lpar1: 1,513,U78AA.001.WZSGVU7-P1-T7,0x21010201,0xc03(USB Controller)
lpar1: 1,512,U78AA.001.WZSGVU7-P1-T9,0x21010200,0x104(RAID Controller)
lpar1: 1/2/2
lpar1: 128.
```

To modify the resource assignment:

```
chvm lpar1 vmcpus=1/2/16 vmmemory=1G/8G/32G add_physlots=0x21010202
```

The output is similar to:

```
lpar1: Success
```

The resource information after modification is similar to:

```
lpar1: Lpar Processor Info:
Curr Processor Min: 1.
Curr Processor Req: 2.
Curr Processor Max: 16.
lpar1: Lpar Memory Info:
Curr Memory Min: 1.00 GB(4 regions).
Curr Memory Req: 8.00 GB(32 regions).
Curr Memory Max: 32.00 GB(128 regions).
lpar1: 1,514,U78AA.001.WZSGVU7-P1-C19,0x21010202,0xffff(Empty Slot)
lpar1: 1,513,U78AA.001.WZSGVU7-P1-T7,0x21010201,0xc03(USB Controller)
lpar1: 1,512,U78AA.001.WZSGVU7-P1-T9,0x21010200,0x104(RAID Controller)
lpar1: 1/2/2
lpar1: 128.
```

Note: The physical I/O resources specified with *add_physlots* will be appended to the specified partition. The physical I/O resources which are not specified but belonged to the partition will not be removed. For more information about *add_physlots*, please refer to lsvm(1)|lsvm.1.

VMware/KVM specific:

```
chvm vm1 -a 8,16 --mem 512 --cpus 2
```

Output is similar to:

```
vm1: node successfully changed
```

zVM specific:

1. To adds a 3390 (ECKD) disk to a virtual machine's directory entry:

```
chvm gpok3 --add3390 POOL1 0101 2G MR
```

Output is similar to:

```
gpok3: Adding disk 0101 to LNX3... Done
```

2. To add a network adapter to a virtual machine's directory entry:

```
chvm gpok3 --addnic 0600 QDIO 3
```

Output is similar to:

```
gpok3: Adding NIC 0900 to LNX3... Done
```

3. To connects a given network adapter to a GuestLAN:

```
chvm gpok3 --connectnic2guestlan 0600 GLAN1 LN1OWNR
```

Output is similar to:

```
gpok3: Connecting NIC 0600 to GuestLan GLAN1 on LN1OWNR... Done
```

4. To connects a given network adapter to a vSwitch:

```
chvm gpok3 --connectnic2vswitch 0600 VSW1
```

Output is similar to:

```
gpok3: Connecting NIC 0600 to vSwitch VSW1 on LNX3... Done
```

5. To removes a minidisk from a virtual machine's directory entry:

```
chvm gpok3 --removedisk 0101
```

Output is similar to:

```
gpok3: Removing disk 0101 on LNX3... Done
```

6. To Removes a network adapter from a virtual machine's directory entry:

```
chvm gpok3 --removenic 0700
```

Output is similar to:

```
gpok3: Removing NIC 0700 on LNX3... Done
```

7. To replaces a virtual machine's directory entry:

```
cat /tmp/dirEntry.txt | chvm gpok3 --replacevs
```

Output is similar to:

```
gpok3: Replacing user entry of LNX3... Done
```

FILES

/opt/xcat/bin/chvm

SEE ALSO

mkvm(1)|mkvm.1, lsvm(1)|lsvm.1, rmvm(1)|rmvm.1

chzone.1

NAME

chzone - Changes a zone defined in the cluster.

SYNOPSIS

chzone <zonename> [**--defaultzone**] [-K] [**-k** *full path to the ssh RSA private key*] [**-a** *noderange* | **-r** *noderange*] [**-g**] [**-f**] [**-s** *yes|no*] [-V]

chzone [**-h** | **-v**]

DESCRIPTION

The **chzone** command is designed to change the definition of a zone previous defined in the cluster. The chzone command is only supported on Linux (No AIX support). The nodes are not updated with the new root ssh keys by chzone. You must run updatenode -k or xdsh -K to the nodes to update the root ssh keys to the new generated zone keys. This will also sync any service nodes with the zone keys, if you have a hierarchical cluster. Note: if any zones in the zone table, there must be one and only one defaultzone. Otherwise, errors will occur.

OPTIONS

-h | --help

> Displays usage information.

-v | --version

> Displays command version and build date.

-k | --sshkeypath *full path to the ssh RSA private key*

> This is the path to the id_rsa key that will be used to build new root's ssh keys for the zone. If -k is used, it will generate the ssh public key from the input ssh RSA private key, and store both in /etc/xcat/sshkeys/<zonename>/.ssh directory.

-K | --genkeys

Using this flag, will generate new ssh RSA private and public keys for the zone into the /etc/xcat/sshkeys/<zonename>/.ssh directory. The nodes are not automatically updated with the new root ssh keys by chzone. You must run updatenode -k or xdsh -K to the nodes to update the root ssh keys to the new generated zone keys. This will also sync any service nodes with the zone keys, if you have a hierarchical cluster.

--default

if –defaultzone is input, then it will set the zone defaultzone attribute to yes. if –defaultzone is input and another zone is currently the default, then the -f flag must be used to force a change to the new defaultzone. If -f flag is not use an error will be returned and no change made. Note: if any zones in the zone table, there must be one and only one defaultzone. Otherwise, errors will occur.

-a | --addnoderange *noderange*

For each node in the noderange, it will set the zonename attribute for that node to the input zonename. If the -g flag is also on the command, then it will add the group name "zonename" to each node in the noderange.

-r | --rmnoderange *noderange*

For each node in the noderange, if the node is a member of the input zone, it will remove the zonename attribute for that node. If any of the nodes in the noderange is not a member of the zone, you will get an error and nothing will be changed. If the -g flag is also on the command, then it will remove the group name "zonename" from each node in the noderange.

-s| --sshbetweennodes yes|no

If -s entered, the zone sshbetweennodes attribute will be set to yes or no based on the input. When this is set to yes, then ssh will be setup to allow passwordless root access between nodes. If no, then root will be prompted for a password when running ssh between the nodes in the zone.

-f | --force

Used with the (–defaultzone) flag to override the current default zone.

-g | --assigngroup

Used with the (-a or -r) flag to add or remove the group zonename for all nodes in the input noderange.

-V | --Verbose

Verbose mode.

Examples

*

To chzone zone1 to the default zone, enter:

chzone *zone1* -****-default -f

*

To generate new root ssh keys for zone2A using the ssh id_rsa private key in /root/.ssh:

chzone *zone2A* -k /root/.ssh

Note: you must use xdsh -K or updatenode -k to update the nodes with the new keys

*

To generate new root ssh keys for zone2A, enter :

chzone *zone2A* -K

Note: you must use xdsh -K or updatenode -k to update the nodes with the new keys

*

To add a new group of nodes (compute3) to zone3 and add zone3 group to the nodes, enter:

chzone *zone3* -a compute3 -g

*

To remove a group of nodes (compute4) from zone4 and remove zone4 group from the nodes, enter:

chzone *zone4* -r compute4 -g

*

To change the sshbetweennodes setting on the zone to not allow passwordless ssh between nodes, enter:

chzone *zone5* -s no

Note: you must use xdsh -K or updatenode -k to update the nodes with this new setting.

Files

/opt/xcat/bin/chzone/

Location of the chzone command.

SEE ALSO

L <mkzone(1)|mkzone.1>,L <rmzone(1)|rmzone.1>,L <xdsh(1)|xdsh.1>, updatenode(1)|updatenode.1

clonevm.1

NAME

clonevm - Create masters from virtual machines and virtual machines from masters.

SYNOPSIS

clonevm noderange [-t <mastertobemade> | -b <master to base vms upon>] -d|–detached -f|–force>

DESCRIPTION

Command to promote a VM's current configuration and storage to a master as well as performing the converse operation of creating VMs based on a master.

By default, attempting to create a master from a running VM will produce an error. The force argument will request that a master be made of the VM anyway.

Also, by default a VM that is used to create a master will be rebased as a thin clone of that master. If the force argument is used to create a master of a powered on vm, this will not be done. Additionally, the detached option can be used to explicitly request that a clone not be tethered to a master image, allowing the clones to not be tied to the health of a master, at the cost of additional storage.

When promoting a VM's current state to master, all rleated virtual disks will be copied and merged with any prerequisite images. A master will not be tethered to other masters.

OPTIONS

-h|--help Display usage message.

-b The master to base the clones upon

-t The target master to copy a single VM's state to

-d Explicitly request that the noderange be untethered from any masters.

-f Force cloning of a powered on VM. Implies -d if the VM is on.

-v|--version Command Version.

-V|--verbose Verbose output.

RETURN VALUE

0: The command completed successfully.

Any other value: An error has occurred.

EXAMPLES

Creating a master named appserver from a node called vm1: *clonevm vm1 -t appserver*

Cleating 30 VMs from a master named appserver: *clonevm vm1-vm30 -b appserver*

FILES

/opt/xcat/bin/clonevm

SEE ALSO

chvm(1)|chvm.1, lsvm(1)|lsvm.1, rmvm(1)|rmvm.1, mkvm(1)|mkvm.1, vmmaster(5)|vmmaster.5

configfpc.1

NAME

configfpc - discover the Fan Power Controllers (FPCs) and configure the FPC interface

SYNOPSIS

configfpc -i *interface*

configfpc -i *interface* **--ip** *default ip address*

configfpc [**-V | --verbose**]

configfpc [**-h | --help | -?**]

DESCRIPTION

configfpc will discover and configure all FPCs that are set to the default IP address. If not supplied the default ip is 192.168.0.100.

The **-i interface** is required to direct **configfpc** to the xCAT MN interface which is on the same VLAN as the FPCs.

There are several bits of information that must be included in the xCAT database before running this command.

You must create the FPC node definitions for all FPCs being discovered including the IP address and switch port information.

The **configfpc** command discovers the FPCs and collects the MAC address. The MAC address is used to relate the FPC to a FPC node using the switch information for this MAC. Once the relationship is discovered the FPC is configured with the FPC node IP settings.

This process is repeated until no more FPCs are discovered.

For more information on xCAT support of NeXtScale and configfpc see the following doc: XCAT_NeXtScale_Clusters

OPTIONS

-i *interface*

> Use this flag to specify which xCAT MN interface (example: eth4) that is connected to the NeXtScale FPCs. This option is required.

--ip *default ip address*

> Use this flag to override the default ip address of 192.168.0.100 with a new address.

-V | --verbose

> Verbose mode

Example

1

> To discover and configure all NeXtScale Fan Power Controllers (FPCs) connected on eth0 interface.
>
> **configfpc -i** *eth0*

2

> To override the default ip address and run in Verbose mode.
>
> **configfpc -i** *eth0* **--ip** *196.68.0.100* **-V**

csm2xcat.1

NAME

csm2xcat - Allows the migration of a CSM database to an xCAT database.

SYNOPSIS

csm2xcat [**--dir** *path*]

csm2xcat [**-h**]

DESCRIPTION

The csm2xcat command must be run on the Management Server of the CSM system that you want to migrate to xCAT. The commmand will build two xCAT stanza files that can update the xCAT database with the chdef command.

Copy the csm2xcat command to the CSM Management Server. Run the command, indicating where you want your stanza files saved with the –dir parameter. Check the stanza files to see if the information is what you want put in the xCAT database. Copy the two stanza files: node.stanza, device.stanza back to your xCAT Management node, and run the chdef command to input into the xCAT database.

OPTIONS

-h Display usage message.

--dir Path to the directory containing the stanza files.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To build xCAT stanza files, enter on the CSM Management Server:

csm2xcat --dir /tmp/mydir

2. To put the data in the xCAT database on the xCAT Management Node:

cat node.stanza | chdef -z

cat device.stanza | chdef -z

FILES

/opt/xcat/share/xcat/tools/csm2xcat

$dir/conversion.log

SEE ALSO

chdef(1)|chdef.1

db2sqlsetup.1

NAME

db2sqlsetup - Sets up the IBM DB2 for xCAT to use.

SYNOPSIS

db2sqlsetup {**-h** | **--help**}

db2sqlsetup {**-v** | **--version**}

db2sqlsetup {**-i** | **--init**}{<-S> | <-C>} [-o|-****-setupODBC] [**-V** | **--verbose**]

db2sqlsetup {**-i** | **--init**}{<-S>} [-N|-****-nostart] [-o|--setupODBC] [**-V** | **--verbose**]

db2sqlsetup {**-o** | **--setupODBC**} {<-S> | <-C>} [-V|-********-verbose]

db2sqlsetup {**-p** | **--passwd**} [<-S> | <-C>]

DESCRIPTION

db2sqlsetup - Sets up the IBM DB2 database for xCAT to use. The db2sqlsetup script is run on the Management Node, after the DB2 Server code has been installed, to setup the DB2 Server (-S). The xcatd daemon will be stopped during migration on the MN. No xCAT commands should be run during the init process, because we will be migrating the xCAT database to DB2 and restarting the xcatd daemon.

The db2sqlsetup script must be run on each Service Node, after the DB2 Client code has been installed, to setup the DB2 Client (-C). There are two postscripts that are provided (db2install and odbcsetup) that will automatically setup you Service Node as a DB2 client.

For full information on the setup of DB2, see Setting_Up_DB2_as_the_xCAT_DB.

When running of db2sqlsetup on the MN: One password must be supplied for the setup, a password for the xcatdb unix id which will be used as the DB2 instance id and database name. The password will be prompted for interactively or can be input with the XCATDB2PW environment variable. The script will create the xcat database instance (xcatdb) in the /var/lib/db2 directory unless overriden by setting the site.databaseloc attribute. This attribute should not be set to the directory that is defined in the installloc attribute and it is recommended that the databaseloc be a new filesystem dedicated to the DB2 database, especially in very large clusters.

When running db2sqlseutp on the SN: Not only will the password for the DB2 instance Id be prompted for and must match the one on the Management Node; but also the hostname or ip address of the Management Node as known by the Service Node must be supplied , unless the XCATDB2SERVER environment variable is set. You can automatically install and setup of DB2 on the SN using the db2install and odbcsetup postscripts and not need to manually run the command. See the full documentation.

Note: On AIX , root must be running ksh and on Linux, bash shell.

OPTIONS

-h|--help

>   Displays the usage message.

-v|--version

>   Displays the release version of the code.

-V|--verbose

>   Displays verbose messages.

-i|--init

>   The init option is used to setup an installed DB2 database on AIX or Linux (p-Series) so that xCAT can
>   use the database. This must be combined with either the -S or -C flag to indicate whether we are setting
>   up the Server or the Client. With the -S flag, it involves creating the xcatdb database, the xcatdb instance
>   id, allowing access to the xcatdb database by the Management Node. It also backs up the current xCAT
>   database and restores it into the newly setup xcatdb DB2 database. It creates the /etc/xcat/cfgloc file to
>   point the xcatd daemon to the DB2 database and restarts the xcatd daemon using the database.

-p|--passwd

>   The password change option is to change the database access password for the DB2 xcatdb database. If -S
>   is input then it will only change the password on the DB2 Server (MN). If -C is input it will only change
>   on the DB2 clients (SN). If neither -S or -C are input with this flag, then it will change both the DB2
>   Server and Clients. When changing the password the xcatd daemon will be stopped and restarted. Any
>   other tools accessing the database should also be stopped before changing and restarted after changing.

-S|-C

>   This options says whether to setup the Server (-S) on the Management Node, or the Client (-C) on the
>   Service Nodes.

-N|--nostart

>   This option with the -S flag will create the database, but will not backup and restore xCAT tables into the
>   database. It will create the cfgloc file such that the next start of xcatd will try and contact the database.
>   This can be used to setup the xCAT DB2 database during or before install.

-o|--setupODBC

>   This option sets up the ODBC /etc/../odbcinst.ini, /etc/../odbc.ini and the .odbc.ini file in roots home
>   directory will be created and initialized to run off the xcatdb DB2 database.

ENVIRONMENT VARIABLES

*

>   XCATDB2INSPATH overrides the default install path for DB2 which is /opt/ibm/db2/V9.7 for Linux and
>   /opt/IBM/db2/V9.7 for AIX.

*

>   DATABASELOC override the where to create the xcat DB2 database, which is /var/lib/db2 by default of
>   taken from the site.databaseloc attribute.

*

>   XCATDB2PW can be set to the password for the xcatdb DB2 instance id so that there will be no prompting
>   for a password when the script is run.

EXAMPLES

*

> To setup DB2 Server for xCAT to run on the DB2 xcatdb database, on the MN:
>
> **db2sqlsetup** *-i -S*

*

> To setup DB2 Client for xCAT to run on the DB2 xcatdb database, on the SN:
>
> **db2sqlsetup** *-i -C*

*

> To setup the ODBC for DB2 xcatdb database access, on the MN :
>
> **db2sqlsetup** *-o -S*

*

> To setup the ODBC for DB2 xcatdb database access, on the SN :
>
> **db2sqlsetup** *-o -C*

*

> To setup the DB2 database but not start xcat running with it:
>
> **db2sqlsetup** *-i -S -N*

*

> To change the DB2 xcatdb password on both the Management and Service Nodes:
>
> **db2sqlsetup** *-p*

dumpxCATdb.1

NAME

dumpxCATdb - dumps the xCAT db tables .

SYNOPSIS

dumpxCATdb [**-a**] [**-V**] [{**-p** | **--path**} *path*]

dumpxCATdb [**-b**] [**-V**] [{**-p** | **--path**} *path*]

dumpxCATdb [**-h** | **--help**] [**-v** | **--version**]

DESCRIPTION

If not using the binary dump option (-b), then the dumpxCATdb command creates .csv files for xCAT database tables and puts them in the directory given by the -p flag. These files can be used by the restorexCATdb command to restore the database. The command will read the list of tables in the site.skiptables attribute and not backup those tables. Supports using XCAT_SKIPTABLES env variable to provide a list of skip tables. The command will never backup TEAL or ISNM tables, except isnm_config. To dump TEAL tables use the documented process for TEAL. For ISNM use tabdump, after using tabprune to get to prune unnecessary records.

If using the binary dump option for the DB2 or postgreSQL database, then the routine will use the Database provide utilites for backup of the entire database.

OPTIONS

-h Display usage message.

-v Command Version.

-V Verbose.

-a All,without this flag the eventlog and auditlog will be skipped.

-b This flag is only used for the DB2 or postgreSQL database. The routine will use the database backup utilities to create a binary backup of the entire database. Note to use this backup on DB2, you will have first had to modify the logging of the database and have taken an offline initial backup. Refer to the xCAT DB2 documentation for more instructions.

-p Path to the directory to dump the database. It will be created, if it does not exist.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To dump the xCAT database into the /tmp/db directory, enter:

dumpxCATdb -p /tmp/db

2. To dump the xCAT database into the /tmp/db directory, including the auditlog and eventlog enter:

dumpxCATdb -a -p /tmp/db

3. To have dumpxCATdb not backup the hosts or passwd table:

chtab key=skiptables site.value="hosts,passwd"

dumpxCATdb -p /tmp/db

4. To have dumpxCATdb not backup the hosts or passwd table:

export XCAT_SKIPTABLES="hosts,passwd"

dumpxCATdb -p /tmp/db

5. To have dumpxCATdb use DB2 utilities to backup the DB2 database:

dumpxCATdb -b -p /install/db2backup

FILES

/opt/xcat/sbin/dumpxCATdb

SEE ALSO

restorexCATdb(1)|restorexCATdb.1

genimage.1

NAME

genimage - Generates a stateless image to be used for a diskless install.

SYNOPSIS

genimage

genimage [**-o** *osver*] [**-a** *arch*] [**-p** *profile*] [**-i** *nodebootif*] [**-n** *nodenetdrivers*] [**--onlyinitrd**] [**-r** *otherifaces*] [**-k** *kernelver*] [**-g** *krpmver*] [**-m** *statelite*] [**-l** *rootlimitsize*] [**--permission** *permission*] [**--interactive**] [**--dryrun**] [**--ignorekernelchk**] [**--noupdate**] *imagename*

genimage -o *osver* [**-a** *arch*] **-p** *profile* **-i** *nodebootif* **-n** *nodenetdrivers* [**--onlyinitrd**] [**-r** *otherifaces*] [**-k** *kernelver*] [**-g** *krpmver*] [**-m** *statelite*] [**-l** *rootlimitsize*] [**--permission** *permission*] [**--interactive**] [**--dryrun**] [**--noupdate**]

genimage [**-h** | **--help** | **-v** | **--version**]

DESCRIPTION

Generates a stateless and a statelite image that can be used to boot xCAT nodes in a diskless mode.

The *imagename* format of the command is recommended. When specified, genimage will use the osimage definition for information to generate this image. Additional options specified on the command line will override any corresponding previous osimage settings, and will be written back to the osimage definition.

If *imagename* is not specified (old method):

> • the default packages included (and excluded) in the image are specified by

/opt/xcat/share/xcat/netboot/<os>/<profile>[.<osver>][.<arch>].pkglist and

/opt/xcat/share/xcat/netboot/<os>/<profile>[.<osver>][.<arch>].exlist.

```
- Additional packages that are not from the os distro can be specified in a
```

/opt/xcat/share/xcat/netboot/<os>/<profile>[.<osver>][.<arch>].otherpkgs.pkglist file.

```
- Customized package list files will override these files and can be specified under /
↪install/custom/netboot/<os> directory.
- The generated image will be put in /install/netboot/<osver>/<arch>/<profile>␣
↪directory.

- osimage definitions will be created in the I<linuximage> and I<osimage> tables. ␣
↪The newly generated image names will have the following format:

    for stateless: <osver>-<arch>-netboot-<profile>

    for statelite: <osver>-<arch>-statelite-<profile>
```

If **genimage** runs on the management node, both the *osimage* table and *linuximage* table will be updated with the given values from the options.

The **genimage** command will generate two initial ramdisks for **stateless** and **statelite**, one is **initrd-stateless.gz**, the other one is **initrd-statelite.gz**.

After your image is generated, you can chroot to the image, install any additional software you would like, or make modifications to files, and then run the following command to prepare the image for deployment.

for stateless: **packimage**

for statelite: **liteimg**

Besides prompting for some paramter values, the **genimage** command takes default guesses for the parameters not specified or not defined in the *osimage* and *linuximage* tables. It also assumes default answers for questions from the yum/zypper command when installing rpms into the image. Please use -********-interactive flag if you want the yum/zypper command to prompt you for the answers.

If **--onlyinitrd** is specified, genimage only regenerates the initrd for a stateless image to be used for a diskless install.

The **genimage** command must be run on a system that is the same architecture and same distro with same major release version as the nodes it will be used on. If the management node is not the same architecture or same distro level, copy the contents of /opt/xcat/share/xcat/netboot/<os> to a system that is the proper architecture, and mount /install from the management node to that system. Then change directory to /opt/xcat/share/xcat/netboot/<os> and run ./genimage.

Parameters

imagename specifies the name of an os image definition to be used. The specification for the image is stored in the *osimage* table and *linuximage* table.

OPTIONS

-a *arch*

> The hardware architecture of this node: x86_64, ppc64, x86, ia64, etc. If omitted, the current hardware architecture will be used.

-o *osver*

> The operating system for the image: fedora8, rhel5, sles10, etc. The OS packages must be in /install/<osver>/<arch> (use copycds(8)|copycds.8).

-p *profile*

> The profile (e.g. compute, service) to use to create the image. This determines what package lists are used from /opt/xcat/share/xcat/netboot/<os> to create the image with. When deploying nodes with this image, the nodes' nodetype.profile attribute must be set to this same value.

-i *nodebootif*

> This argument is now optional, and allows you to specify the network boot interface to be configured in the image (e.g. eth0). If not specified, the interface will be determined and configured during the network boot process.

-n *nodenetdrivers*

> This argument is now optional, and allows you to specify the driver modules needed for the network interface(s) on your stateless nodes. If you do not specify this option, the default is to include all recent IBM xSeries network drivers.

If specified, *nodenetdrivers* should be a comma separated list of network drivers to be used by the stateless nodes (Ie.: -n tg3,e1000). Note that the drivers will be loaded in the order that you list them, which may prove important in some cases.

-l *rootlimit*

The maximum size allowed for the root file system in the image. Specify in bytes, or can append k, m, or g.

--onlyinitrd

Regenerates the initrd for a stateless image to be used for a diskless install.

Regenerates the initrd that is part of a stateless/statelite image that is used to boot xCAT nodes in a stateless/stateli te mode. The generated initrd will be put in /install/netboot/<OS>/<arch>/<profile>.

The **genimage --onlyinitrd** command will generate two initial ramdisks, one is **initrd-statelite.gz** for **statelite** mode, the other one is **initrd-stateless.gz** for **stateless** mode.

--permission *permission*

The mount permission of **/.statelite** directory for **statelite** mode, which is only used for **statelite** mode, and the default permission is 755.

-r *otherifaces*

Other network interfaces (e.g. eth1) in the image that should be configured via DHCP.

-k *kernelver*

Use this flag if you want to use a specific version of the kernel in the image. Defaults to the first kernel found in the install image.

-g *krpmver*

Use this flag to specify the rpm version for kernel packages in the image. It must be present if -k flag is specified in the command for SLES. Generally, the value of -g is the part after **linux-** and before **.rpm** in a kernel rpm name.

-m statelite

This flag is for Ubuntu, Debian and Fedora12 only. Use this flag to specify if you want to generate statelite image. The default is to generate stateless image for these three operating systems. For others, this flag is invalid because both stateless and statelite images will be generated with this command.

--interactive

This flag allows the user to answer questions from yum/zypper command when installing rpms into the image. If it is not specified, '-y' will be passed to the yum command and '--non-interactive --no-gpg-checks' will be passed to the zypper command as default answers.

--dryrun

This flag shows the underlying call to the os specific genimage function. The user can copy and the paste the output to run the command on another machine that does not have xCAT installed.

-t *tmplimit*

(Deprecated) This flag allows the user to setup the /tmp and the /var/tmp file system sizes. This flag is no longer supported. You can overwrite any file system size using the .postinstall script where you can create a new /etc/fstab file.

--ignorekernelchk

Skip the kernel version checking when injecting drivers from osimage.driverupdatesrc. That means all drivers from osimage.driverupdatesrc will be injected to initrd for the specific target kernel.

--noupdate

> This flag allows the user to bypass automatic package updating when installing other packages.

-v|--version

> Display version.

-h|--help

> Display usage message.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1 To prompt the user for inputs:

```
genimage
```

2 To generate an image using information from an osimage definition:

```
genimage myimagename
```

3 To run genimage in test mode without actually generating an image:

```
genimage --dryrun  myimagename
```

4 To generate an image and have yum/zypper prompt for responses:

```
genimage myimagename --interactive
```

5 To generate an image, replacing some values in the osimage definition:

```
genimage -i eth0 -n tg3 myimagename
```

6 (old method) To generate a fedora8 image for a compute node architecture x86_64 and place it in the /install/netboot/fedora8/x86_64/compute/rootimg directory:

```
genimage -i eth0 -o fedora8 -p compute
```

7 (old method)

```
genimage -i eth0 -r eth1,eth2 -n tg3,bnx2 -o centos5.1 -p compute
```

8 (old method)

```
genimage -i eth0 -n tg3,bnx2 -o sles11 -p compute --interactive
```

9 (old method)

```
genimage -i eth0 -n igb,e1000e,e1000,bnx2,tg3 -o centos5.4 -p nfsroot --
↪permission 777
```

10 (old method) To regenerate the initrd for a fedora8 image for a compute node architecture x86_64 and place it in the /install/netboot/fedora8/x86_64/compute/rootimg directory:

```
cd /opt/xcat/share/xcat/netboot/fedora
./genimage --onlyinitrd -i eth0 -n tg3,bnx2 -o fedora8 -p compute
```

FILES

/opt/xcat/bin/genimage

/opt/xcat/share/xcat/netboot/<OS>/genimage

SEE ALSO

packimage(1)|packimage.1, liteimg(1)|liteimg.1

geninitrd.1

NAME

geninitrd - Generate an initrd (initial ramfs) which to be used for statefull install or stateless netboot.

SYNOPSIS

geninitrd *imagename* [**--ignorekernelchk**]

geninitrd [**-h** | **--help**]

DESCRIPTION

Generate the initrd for the osimage: **imagename** which is an xCAT object of *osimage* type.

Diskfull Osimage

If the **imagename** is a statefull one (The provmethod attribute for the osimage is 'install'), this command is used to rebuild the initrd to inject the new drivers from driver rpms or 'update distro' and copy the rebuilt initrd and new kernel (If there's new kernel in 'update distro') to the directory */tftpboot/xcat/<imagename>*.

If the initrd has been rebuilt by geninitrd, when run nodeset, the *–noupdateinitrd* option should be used to skip the rebuilding of initrd to improve the performance.

Three attributes of osimage object can be used to specify the Driver RPM location and Driver names for injecting new drviers to initrd.

netdrivers - comma separated driver names that need to be injected to the initrd. The postfix '.ko' can be ignored. The netdrivers attribute must be set to specify the new driver list. If you want to load all the drivers from the driver rpms, using the keyword allupdate.

driverupdatesrc - comma separated driver rpm packages (full path should be specified)

osupdatename - comma separated 'osdistroupdate' object. Each 'osdistroupdate' object specifies a Linux distro update. When run geninitrd, 'kernel-*.rpm' will be searched from osdistroupdate.dirpath to get all the rpm packages and then search the drivers from the rpm packages.

Refer to the doc: Using_Linux_Driver_Update_Disk

Stateless Osimage

If the **imagename** is a stateless one (The provmethod attribute for the osimage is 'netboot'), this command is used to generate the initrd from the rootimg which generated by 'genimage' command. So the 'genimage' must be run once before running the geninitrd command.

Two attributes of osimage object can be used to specify the Driver RPM location and Driver names for injecting new drviers to initrd.

netdrivers - comma separated driver names that need to be injected to the initrd. The postfix '.ko' can be ignored. The netdrivers attribute must be set to specify the new driver list. If you want to load all the drivers from the driver rpms, using the keyword allupdate.

driverupdatesrc - comma separated driver rpm packages (full path should be specified)

Parameters

imagename specifies the name of an os image definition to be used. The specification for the image is storted in the *osimage* table and *linuximage* table.

--ignorekernelchk

> Skip the kernel version checking when injecting drivers from osimage.driverupdatesrc. That means all drivers from osimage.driverupdatesrc will be injected to initrd for the specific target kernel.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1 To generate initrd for the osimage **myimagename**:

```
geninitrd myimagename
```

FILES

/opt/xcat/bin/geninitrd

/opt/xcat/bin/genimage

/opt/xcat/share/xcat/netboot/<OS>/genimage

SEE ALSO

geninitrd(1)|geninitrd.1, genimage(1)|genimage.1

getmacs.1

NAME

getmacs - Collects node MAC address.

SYNOPSIS

Common:

getmacs [**-h**| **--help** | **-v**| **--version**]

PPC specific:

getmacs *noderange* [**-F** *filter*]

getmacs *noderange* [**-M**]

getmacs *noderange* [**-V**| **--verbose**] [**-f**] [**-d**] [**--arp**] | [**-D** [**-S** *server*] [**-G** *gateway*] [**-C** *client*] [**-o**]]

blade specific:

getmacs *noderange* [**-V**| **--verbose**] [**-d**] [**--arp**] [**-i** *ethN*|*enN*]

DESCRIPTION

The getmacs command collects MAC address from a single or range of nodes. Note that on AIX systems, the returned MAC address is not colon-seperated (for example 8ee2245cf004), while on Linux systems the MAC address is colon-seperated (for example 8e:e2:24:5c:f0:04). If no ping test performed, getmacs writes the first adapter MAC to the xCAT database. If ping test performed, getmacs will write the first successfully pinged MAC to xCAT database.

For PPC (using Direct FSP Management) specific:

Note: If network adapters are physically assigned to LPARs, getmacs cannot read the MAC addresses unless perform ping test, since there is no HMC command to read them and getmacs has to login to open formware. And if the LPARs has never been activated before, getmacs need to be performed with the option "**-D**" to get theirs MAC addresses.

For PPC (using HMC) specific:

Note: The option "**-D**" **must** be used to get MAC addresses of LPARs.

For IBM Flex Compute Node (Compute Node for short) specific:

Note: If "**-d**" is specified, all the MAC of the blades will be displayed. If no option specified, the first MAC address of the blade will be written to mac table.

OPTIONS

--arp

Read MAC address with ARP protocal.

-C

Specify the IP address of the partition for ping test. The default is to read from xCAT database if no **-C** specified.

-d

Display MAC only. The default is to write the first valid adapter MAC to the xCAT database.

-D

Perform ping test. Please be aware that in this way, the lpars will be reset.

-f

Force immediate shutdown of the partition.This flag must be used with -D flag.

-F

Specify filters to select the correct adapter. Acceptable filters are Type, MAC_Address, Phys_Port_Loc, Adapter, Port_Group, Phys_Port, Logical_Port, VLan, VSwitch, Curr_Conn_Speed.

-G

Gateway IP address of the partition. The default is to read from xCAT database if no **-G** specified.

-h

Display usage message.

-M

Return multiple MAC addresses for the same adapter or port, if available from the hardware. For some network adapters (e.g. HFI) the MAC can change when there are some recoverable internal errors. In this case, the hardware can return several MACs that the adapter can potentially have, so that xCAT can put all of them in DHCP. This allows successful booting, even after a MAC change, but on Linux at this time, it can also cause duplicate IP addresses, so it is currently not recommended on Linux. By default (without this flag), only a single MAC address is returned for each adapter.

-o

Read MAC address when the lpar is in openfirmware state. This option mush be used with [**-D**] option to perform ping test. Before use **-o**, the lpar must be in openfirmware state.

-S

The IP address of the machine to ping. The default is to read from xCAT databse if no **-S** specified.

-v

Command Version.

-V

Verbose output.

-i

Specify the interface whose mac address will be collected and written into mac table. If 4 mac addresses are returned by option '-d', they all are the mac addresses of the blade. The N can start from 0(map to the eth0 of the blade) to 3. If 5 mac addresses are returned, the 1st mac address must be the mac address of the blade's FSP, so the N will start from 1(map to the eth0 of the blade) to 4.

RETURN VALUE

```
0 The command completed successfully.

1 An error has occurred.
```

EXAMPLES

1. To retrieve the MAC address for the HMC-managed partition lpar4 and write the first valid adapter MAC to the xCAT database, enter:

```
getmacs lpar4
```

Output is similar to:

```
lpar4:
#Type  MAC_Address  Phys_Port_Loc  Adapter  Port_Group  Phys_Port  Logical_Port  VLan
↪ VSwitch  Curr_Conn_Speed
hea  7607DFB07F02  N/A  N/A  N/A  N/A  N/A  1  ETHERNET0  N/A
ent U78A1.001.99203B5-P1-T6   00145eb55788 /lhea@23c00614/ethernet@23e00514
↪unsuccessful physical
```

2. To retrieve the MAC address with ARP protocal:

```
getmacs lpar4 --arp
```

Output is similar to:

```
lpar4:
#IP            MAC_Address
192.168.0.10   00145eb55788
```

3. To retrieve the MAC address for the HMC-managed partition lpar4 and display the result only, enter:

```
getmacs lpar4 -d
```

Output is similar to:

```
lpar4:
#Type  MAC_Address  Phys_Port_Loc  Adapter  Port_Group  Phys_Port  Logical_Port  VLan
↪ VSwitch  Curr_Conn_Speed
hea  7607DFB07F02  N/A  N/A  N/A  N/A  N/A  1  ETHERNET0  N/A
ent U78A1.001.99203B5-P1-T6   00145eb55788 /lhea@23c00614/ethernet@23e00514
↪unsuccessful physical
```

4. To retrieve the MAC address for the HMC-managed partition lpar4 with filter Type=hea,VSwitch=ETHERNET0.

```
getmacs lpar4 -F Type=hea,VSwitch=ETHERNET0
```

Output is similar to:

```
lpar4:
#Type  MAC_Address  Phys_Port_Loc  Adapter  Port_Group  Phys_Port  Logical_Port  VLan
↪ VSwitch  Curr_Conn_Speed
hea  7607DFB07F02  N/A  N/A  N/A  N/A  N/A  1  ETHERNET0  N/A
```

5. To retrieve the MAC address while performing a ping test for the HMC-managed partition lpar4 and display the result only, enter:

```
getmacs lpar4 -d -D -S 9.3.6.49 -G 9.3.6.1 -C 9.3.6.234
```

Output is similar to:

```
lpar4:
#Type  Location Code   MAC Address      Full Path Name  Ping Result
ent U9133.55A.10B7D1G-V12-C4-T1 8e:e2:24:5c:f0:04 /vdevice/l-lan@30000004 successful␣
↪virtual
```

6. To retrieve the MAC address for Power 775 LPAR using Direct FSP Management without ping test and display
   the result only, enter:

```
getmacs lpar4 -d
```

Output is similar to:

```
lpar4:
#Type  Phys_Port_Loc  MAC_Address  Adapter  Port_Group  Phys_Port  Logical_Port  VLan␣
↪ VSwitch  Curr_Conn_Speed
HFI  N/A  02:00:02:00:00:04  N/A  N/A  N/A  N/A  N/A  N/A  N/A
```

7. To retrieve multiple MAC addresses from Power 775 HFI network adapter using Direct FSP Management, enter:

```
getmacs lpar4 -M
```

Output is similar to:

```
lpar4:
#Type  Phys_Port_Loc  MAC_Address  Adapter  Port_Group  Phys_Port  Logical_Port  VLan␣
↪ VSwitch  Curr_Conn_Speed
HFI  N/A  02:00:02:00:00:04|02:00:02:00:00:05|02:00:02:00:00:06  N/A  N/A  N/A  N/A ␣
↪N/A  N/A  N/A
```

FILES

/opt/xcat/bin/getmacs

SEE ALSO

makedhcp(8)|makedhcp.8

getslnodes.1

NAME

getslnodes - queries your SoftLayer account and gets attributes for each server.

SYNOPSIS

getslnodes [**-v** | **--verbose**] [*hostname-match*]

getslnodes [**-?** | **-h** | **--help**]

DESCRIPTION

The **getslnodes** command queries your SoftLayer account and gets attributes for each server. The attributes can be piped to 'mkdef -z' to define the nodes in the xCAT DB so that xCAT can manage them.

Before using this command, you must download and install the SoftLayer API perl module. For example:

```
cd /usr/local/lib
git clone https://github.com/softlayer/softlayer-api-perl-client.git
```

You also need to follow these directions to get your SoftLayer API key: http://knowledgelayer.softlayer.com/procedure/retrieve-your-api-key

getslnodes requires a .slconfig file in your home directory that contains your SoftLayer userid, API key, and location of the SoftLayer API perl module, in attr=val format. For example:

```
# Config file used by the xcat cmd getslnodes
userid = joe_smith
apikey = 1234567890abcdef1234567890abcdef1234567890abcdef
apidir = /usr/local/lib/softlayer-api-perl-client
```

OPTIONS

-?|-h|--help

> Display usage message.

-v|--version

> Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. Display information about all of the nodes in your SoftLayer account:

   ```
   getslnodes
   ```

2. Display information about all of the nodes whose hostname starts with foo:

   ```
   getslnodes foo
   ```

3. Create xCAT node defintions in the xCAT DB for all of the nodes in your SoftLayer account:

   ```
   getslnodes | mkdef -z
   ```

FILES

/opt/xcat/bin/getslnodes

SEE ALSO

pushinitrd(1)|pushinitrd.1

gettab.1

NAME

gettab - select table rows, based on attribute criteria, and display specific attributes.

SYNOPSIS

gettab [**-H** | **--with-fieldname**] *key=value,... table.attribute ...*

gettab [**-?** | **-h** | **--help**]

DESCRIPTION

The **gettab** command uses the specified key values to select a row in each of the tables requested. For each selected row, the specified attributes are displayed. The **gettab** command can be used instead of **nodels** for tables that are not keyed by nodename (e.g. the **site** table), or to select rows based on an attribute value other than nodename.

OPTIONS

-H|--with-fieldname

> Always display table.attribute name next to result. By default, this is done only if more than one table.attribute is requested.

-?|-h|--help

> Display usage message.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

*

> To display setting for **master** (management node) in the site table:

> **gettab -H** *key=master site.value*

> The output would be similar to:

```
site.value: mgmtnode.cluster.com
```

*

To display the first node or group name that has **mgt** set to **blade** in the nodehm table:

gettab *mgt=blade nodehm.node*

The output would be similar to:

```
blades
```

FILES

/opt/xcat/bin/gettab

SEE ALSO

nodels(1)|nodels.1, chtab(8)|chtab.8, tabdump(8)|tabdump.8

getxcatdocs.1

NAME

getxcatdocs - downloads the xCAT documentation and converts to HTML and PDF

SYNOPSIS

getxcatdocs [**-?** | **-h** | **--help**] **getxcatdocs** [**-v** | **--verbose**] [*destination-dir*] **getxcatdocs** [**-v** | **--verbose**] [**-c** | **--continue**] [**-d** | **--doc** *single_doc*] [*destination-dir*]

DESCRIPTION

The **getxcatdocs** command downloads the xCAT documentation from the wiki and converts it to both HTML and PDF. This enables reading the documentation when you do not have internet access. Note that this command does not download/convert the entire xCAT wiki - only the "official" xCAT documentation linked from http://sourceforge.net/p/xcat/wiki/XCAT_Documentation.

If *destination-dir* is specified, **getxcatdocs** will put the converted documentation in that directory, in 3 sub-directories: html, pdf, images. Otherwise, it will put it in the current directory (in the same three sub-directories).

If **--doc** *single_doc* is specified, only that one wiki page will be downloaded and converted.

getxcatdocs uses curl to run the Allura wiki API to download the document markdown text, and Pandoc with LaTex them to PDF. You must have all of these functions installed to run **getxcatdocs**. See: http://sourceforge.net/p/xcat/wiki/Editing_and_Downloading_xCAT_Documentation/#converting-wiki-pages-to-html-and-pdfs

Limitations:

*

This command does not run on AIX or Windows.

OPTIONS

-?|-h|--help

Display usage message.

-v|--verbose

Run the command in verbose mode.

-c|--continue

If a previous run of this command failed (which often happens if you lose your network connection), continue processing using files already downloaded to your markdown directory.

-d|--doc *single_doc*

Run this command for a single document only. If you get errors about Official-xcat-doc.png not found, either download this image directly from http://sourceforge.net/p/xcat/wiki/XCAT_Documentation/attachment/Official-xcat-doc.png or run **getxcatdocs -d XCAT_Documentation** first.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To download/convert the documentation and put it in ~/tmp:

```
getxcatdocs ~/tmp
```

FILES

/opt/xcat/bin/getxcatdocs

groupfiles4dsh.1

NAME

groupfiles4dsh - Builds a directory of files for each defined nodegroup in xCAT.

SYNOPSIS

groupfiles4dsh [{**-p** | **--path**} *path*]

groupfiles4dsh [**-h** | **--help**] [**-v** | **--version**]

DESCRIPTION

This tool will build a directory of files, one for each defined nodegroup in xCAT. The file will be named the nodegroup name and contain a list of nodes that belong to the nodegroup. The file can be used as input to the AIX dsh command. The purpose of this tool is to allow backward compatiblity with scripts that were created using the AIX or CSM dsh command

Reference: man dsh.

OPTIONS

-h Display usage message.

-v Command Version.

-p Path to the directory to create the nodegroup files (must exist).

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To create the nodegroup files in directory /tmp/nodegroupfiles, enter:

groupfiles4dsh -p /tmp/nodegroupfiles

To use with dsh:

```
export DSH_CONTEXT=DSH  ( default unless CSM is installed)
export DSH_NODE_RSH=/bin/ssh   (default is rsh)
export DSH_NODEGROUP_PATH= /tmp/nodegroupfiles

dsh  -N all  date   (where all is a group defined in xCAT)
dsh -a date  (will look in all nodegroupfiles and build a list of all nodes)
```

FILES

/opt/xcat/share/xcat/tools/groupfiles4dsh

SEE ALSO

xdsh(1)|xdsh.1

imgcapture.1

NAME

imgcapture - Captures an image from a Linux diskful node and create a diskless or diskful image on the management node.

SYNOPSIS

imgcapture node **-t | --type** diskless [**-p | --profile** *profile*] [**-o | --osimage** *osimage*] [**-i** *nodebootif*] [**-n** *nodenetdrivers*] [**-V | --verbose**]

imgcapture node **-t | --type** sysclone **-o | --osimage** *osimage* [**-V | --verbose**]

imgcapture [**-h | --help**] | [**-v | --version**]

DESCRIPTION

The **imgcapture** command will capture an image from one running diskful Linux node and create a diskless or diskful image for later use.

The **node** should be one diskful Linux node, managed by the xCAT MN, and the remote shell between MN and the **node** should have been configured. AIX is not supported.

The **imgcapture** command supports two image types: **diskless** and **sysclone**. For the **diskless** type, it will capture an image from one running diskful Linux node, prepares the rootimg directory, kernel and initial rmadisks for the **liteimg/packimage** command to generate the statelite/stateless rootimg. For the **sysclone** type, it will capture an image from one running diskful Linux node, create an osimage which can be used to clone other diskful Linux nodes.

The **diskless** type:

The *osimage* is pre-defined, the attributes of osimage will be used to capture and prepare the root image. The **osver**, **arch** and **profile** attributes for the stateless/statelite image to be created are duplicated from the **node**'s attribute. If the **-p|--profile** *profile* option is specified, the image will be created under "/<*install-root*>/netboot/<osver>/<arch>/<*profile*>/rootimg".

The default files/directories excluded in the image are specified by /opt/xcat/share/xcat/netboot/<os>/<*profile*>.<osver>.<arch>.imgcaptu also, you can put your customized file (<*profile*>.<osver>.<arch>.imgcapture.exlist) to /in-stall/custom/netboot/<osplatform>. The directories in the default *.imgcapture.exlist* file are necessary to capture image from the diskful Linux node managed by xCAT, please don't remove it.

The image captured will be extracted into the /<*installroot*>/netboot/**<osver>**/**<arch>**/**<profile>**/rootimg directory.

After the **imgcapture** command returns without any errors, you can customize the rootimg and run the **liteimg/packimage** command with the options you want.

The **sysclone** type:

xCAT leverages the Open Source Tool - Systemimager to capture the osimage from the **node**, and put it into /<*install-root*>/**sysclone/images** directory.

The **imgcapture** command will create the *osimage* definition after the image is captured successfully, you can use this osimage and **nodeset** command to clone diskful nodes.

OPTIONS

-t | --type

> Specify the osimage type you want to capture, two types are supported: diskless and sysclone.

-p|--profile *profile*

> Assign *profile* as the profile of the image to be created.

-o|--osimage *osimage*

> The osimage name.

-i *nodebootif*

> The network interface the diskless node will boot over (e.g. eth0), which is used by the **genimage** command to generate initial ramdisks.

> This is optional.

-n *nodenetdrivers*

> The driver modules needed for the network interface, which is used by the **genimage** command to generate initial ramdisks.

> This is optional. By default, the **genimage** command can provide drivers for the following network interfaces:

> For x86 or x86_64 platform:

```
tg3 bnx2 bnx2x e1000 e1000e igb m1x_en
```

> For ppc64 platform:

```
e1000 e1000e igb ibmveth ehea
```

> For S390x:

```
qdio ccwgroup
```

> If the network interface is not in the above list, you'd better specify the driver modules with this option.

-h|--help

> Display the usage message.

-v|--version

> Display the version.

-V|--verbose

> Verbose output.

RETRUN VALUE

0 The command completed sucessfully.

1 An error has occurred.

EXAMPLES

node1 is one diskful Linux node, which is managed by xCAT.

1. In order to capture and prepare the diskless root image, run the following command:

imgcapture node1 -t diskless

2. In order to capture and prepare the diskless root image with **hpc** as profile, run the command:

imgcapture node1 -t diskless -p hpc

3. In order to capture and prepare the diskless root image: its profile is **hpc**, and the network interface the diskless node will boot over is **eth0**, the driver modules for this network interface is **e1000e**.

imgcapture node1 -t diskless -p hpc -i eth0 -n e1000e

4. There's one pre-defined *osimage*. In order to capture and prepare the diskless root image for *osimage*, run the command:

imgcapture node1 -t diskless -o osimage

5. In order to capture the diskful image from **node1** and create the *osimage* **img1**, run the command:

imgcapture node1 -t sysclone -o img1

FILES

/opt/xcat/bin/imgcapture

SEE ALSO

genimage(1)|genimage.1,   imgimport(1)|imgimport.1,   imgexport(1)|imgexport.1,   packimage(1)|packimage.1,
liteimg(1)|liteimg.1, nodeset(8)|nodeset.8

imgexport.1

NAME

imgexport - Exports an xCAT image.

SYNOPSIS

imgexport [-h| --help]

imgexport image_name [destination] [[-e|--extra file:dir] ...] [-p|--postscripts node_name] [-v|--verbose]

DESCRIPTION

The imgexport command will export an image that is being used by xCAT. To export images, you must have the images defined in the *osimage* table. All the columns in the *osimage* and *linuximage* tables will be exported. If kits are used in statefull or stateless images, *kit*, *kitcomponent* and *kitrepo* tables will be exported. In addition, the following files will also be exported.

For statefull: x.pkglist x.otherpkgs.pkglist x.tmpl x.synclist kits related files

For stateless: kernel initrd.gz rootimg.gz x.pkglist x.otherpkgs.pkglist x.synclist x.postinstall x.exlist kits related files

For statelite: kernel initrd.gz root image tree x.pkglist x.synclist x.otherpkgs.pkglist x.postinstall x.exlist

where x is the name of the profile.

Any files specified by the -e flag will also be exported. If -p flag is specified, the names of the postscripts and the postbootscripts for the given node will be exported. The postscripts themsleves need to be manualy exported using -e flag.

For statelite, the litefile table settings for the image will also be exported. The litetree and statelite tables are not exported.

OPTIONS

-e|--extra *srcfile:destdir* Pack up extra files. If *destdir* is omitted, the destination directory will be the same as the source directory.

-h|--help Display usage message.

-p|--postscripts *node_name* Get the names of the postscripts and postbootscripts for the given node and pack them into the image.

-v|--verbose Verbose output.

image_name The name of the image. Use *lsdef -t* osimage to find out all the image names.

destination The output bundle file name.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. Simplest way to export an image. If there is an image in the osimage table named 'foo', then run:

imgexport foo

foo.tgz will be built in the current working directory. Make sure that you have enough space in the directory that you are in to run imgexport if you have a big image to tar up.

2. To include extra files with your image:

imgexport Default_Stateless_1265981465 foo.tgz -e /install/postscripts/myscript1 -e /tmp/mydir:/usr/mydir

In addition to all the default files, this will export */install/postscripts/myscript1* and the whole directory */tmp/dir* into the file called foo.tgz. And when imgimport is called */install/postscripts/myscript1* will be copied into the same directory and */tmp/mydir* will be copied to */usr/mydir*.

3. To include postscript with your image:

imgexport Default_Stateless_1265981465 foo.tgz -p node1 -e /install/postscripts/myscript1

The *postscripts* and the *postbootscripts* names specified in the *postscripts* table for node1 will be exported into the image. The postscript *myscript1* will also be exported.

FILES

/opt/xcat/bin/imgexport

SEE ALSO

imgimport(1)|imgimport.1

imgimport.1

NAME

imgimport - Imports an xCAT image or configuration file into the xCAT tables so that you can immediately begin deploying with it.

SYNOPSIS

imgimport [-h|--help]

imgimport bundle_file_name [-p|-****-postscripts nodelist] [-f|--profile new_profile] [-v|-****-verbose]>

DESCRIPTION

The imgimport command will import an image that has been exported by *imgexport* from xCAT. This is the easiest way to transfer/backup/, change or share images created by xCAT whether they be stateless or stateful. The bundle file will be unpacked in the current working directory. The xCAT configuration such as *osimage* and *linuximage* tables will then be updated.

For statefull, the following files will be copied to the appropriate directories. x.pkglist x.otherpkgs.pkglist x.tmpl x.synclist kits related files

For stateless, the following files will be copied to the appropriate directories. kernel initrd.gz rootimg.gz x.pkglist x.otherpkgs.pkglist x.synclist x.postinstall x.exlist kits related files

For statelite, the following files will be copied to the appropriate directories. kernel initrd.gz root image tree x.pkglist x.synclist x.otherpkgs.pkglist x.postinstall x.exlist

where x is the profile name.

Any extra files, included by –extra flag in the imgexport command, will also be copied to the appropriate directories.

For statelite, the litefile table will be updated for the image. The litetree and statelite tables are not imported.

If -p flag is specified, the *postscripts* table will be updated with the postscripts and the postbootscripts names from the image for the nodes given by this flag.

If -f flag is not specified, all the files will be copied to the same directories as the source. If it is specified, the old profile name x will be changed to the new and the files will be copied to the appropriate directores for the new profiles. For example, */opt/xcat/share/xcat/netboot/sles/x.pkglist* will be copied to */install/custom/netboot/sles/compute_new.pkglist* and */install/netboot/sles11/ppc64/x/kernel* will be copied to */install/netboot/sles11/ppc64/compute_new/kernel*. This flag is commonly used when you want to copy the image on the same xCAT mn so you can make modification on the new one.

After this command, you can run the *nodeset* command and then start deploying the nodes. You can also choose to modify the files and run the following commands before the node depolyment.

For statefull: nodeset

For stateless: genimage packimage nodeset

For statelite genimage liteimg nodeset

OPTIONS

-f|--profile *new_prof* Import the image with a new profile name.

-h|--help Display usage message.

-p|--postscripts *nodelist* Import the postscripts. The postscripts contained in the image will be set in the postscripts table for *nodelist*.

-v|--verbose Verbose output.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. Simplest way to import an image. If there is a bundle file named 'foo.gz', then run:

imgimport foo.gz

2. Import the image with postscript names.

imgimport foo.gz -p node1,node2

The *postscripts* table will be updated with the name of the *postscripts* and the *postbootscripts* for node1 and node2.

3. Import the image with a new profile name

imgimport foo.gz -f compute_test

FILES

/opt/xcat/bin/imgimport

SEE ALSO

imgexport(1)|imgexport.1

liteimg.1

NAME

liteimg - Modify statelite image by creating a series of links.

SYNOPSIS

liteimg [-h| –help]

liteimg [-v| –version]

liteimg [-o OS] [-p profile] [-a architecture] [-t rootfstype]

liteimg imagename

DESCRIPTION

This command modifies the statelite image by creating a series of links. It creates 2 levels of indirection so that files can be modified while in their image state as well as during runtime. For example, a file like <$imgroot>/etc/ntp.conf will have the following operations done to it:

* mkdir -p $imgroot/.default/etc*

* mkdir -p $imgroot/.statelite/tmpfs/etc*

* mv $imgroot/etc/ntp.conf $imgroot/.default/etc*

* cd $imgroot/.statelite/tmpfs/etc*

* ln -sf ../../../.default/etc/ntp.conf .*

* cd $imgroot/etc*

* ln -sf ../.statelite/tmpfs/etc/ntp.conf .*

When finished, the original file will reside in *$imgroot/.default/etc/ntp.conf*. *$imgroot/etc/ntp.conf* will link to *$imgroot/.statelite/tmpfs/etc/ntp.conf* which will in turn link to *$imgroot/.default/etc/ntp.conf*

Note: If you make any changes to your litefile table after running liteimg then you will need to rerun liteimg again.

Parameters

imagename specifies the name of a os image definition to be used. The specification for the image is storted in the *osimage* table and *linuximage* table.

OPTIONS

-h Display usage message.

-v Command Version.

-o Operating system (fedora8, rhel5, sles10,etc)

-p Profile (compute,service)

-a Architecture (ppc64,x86_64,etc)

-t The type of rootfs for the image (nfs, ramdisk). It is optional, nfs is used to be the rootfs type when **-t** is not specified.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To lite a fedora8 image for a compute node architecture x86_64 enter:

liteimg -o fedora8 -p compute -a x86_64

FILES

/opt/xcat/bin/

NOTES

This command is part of the xCAT software product.

SEE ALSO

genimage(1)|genimage.1

lsdef.1

NAME

lsdef - Use this command to list xCAT data object definitions.

SYNOPSIS

lsdef [**-h** | **--help**] [**-t** *object-types*] [**-i** *attr-list*]

lsdef [**-V** | **--verbose**] [**-l** | **--long**] [**-s** | **--short**] [**-a** | **--all**] [**-S**] [**-t** *object-types*] [**-o** *object-names*] [**-z** | **--stanza**] [**-i** *attr-list*] [**-c** | **--compress**] [**--osimage**] [**--nics**] [[**-w** *attr==val*] [**-w** *attr=~val*] ...] [*noderange*]

DESCRIPTION

This command is used to display xCAT object definitions which are stored in the xCAT database.

OPTIONS

-a|--all

> Display all definitions. For performance consideration, the auditlog and eventlog objects will not be listed. To list auditlog or eventlog objects, use lsdef -t auditlog or lsdef -t eventlog instead.

-c|--compress

> Display information in compressed mode, each output line has format "<object name>: <data>". The output can be passed to command xcoll or xdshbak for formatted output. The -c flag must be used with -i flag.

-h|--help

> Display usage message.

-i *attr-list*

> Comma separated list of attribute names to display.

-l|--long

> List the complete object definition.

-s|--short

> Only list the object names.

-S

> List all the hidden nodes (FSP/BPA nodes) with other ones.

noderange

> A set of comma delimited node names and/or group names. See the "noderange" man page for details on supported formats.

-o *object-names*

> A set of comma delimited object names.

--osimage

> Show all the osimage information for the node.

--nics

> Show the nics configuration information for the node.

-t *object-types*

> A set of comma delimited object types. Use the help option to get a list of valid objects.

-V|--verbose

> Verbose mode.

-w *attr==val* **-w** *attr=~val* ...

> Use one or multiple -w flags to specify the selection string that can be used to select objects. The operators ==, !=, =~ and !~ are available. Use the help option to get a list of valid attributes for each object type.

> **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.

Note: if the "val" fields includes spaces or any other characters that will be parsed by shell, the "attr<operator>val" needs to be quoted. If the operator is "!~", the "attr<operator>val" needs to be quoted using single quote.

-z|--stanza

Display output in stanza format. See the xcatstanzafile man page for details on using xCAT stanza files.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To display a description of all the valid attributes that could be used when defining an xCAT node.

```
lsdef -t node -h
```

2. To get a list of all the objects that have been defined.

```
lsdef
  OR
lsdef -a
```

3. To get all the attributes of the node1

```
lsdef node1
  OR
lsdef -t node node1
  OR
lsdef -t node -o node1
```

4. To get the object name of node1 instead of all the attributes

```
lsdef -s node1
```

5. To get a list of all the network definitions.

```
lsdef -t network
```

6. To get a complete listing of all network definitions.

```
lsdef -l -t network
```

7. To list the whole xCAT database and write it to a stanza file. (backup database)

```
lsdef -a -l -z > mydbstanzafile
```

8. To list the MAC and install adapter name for each node.

```
lsdef -t node -i mac,installnic
```

9. To list an osimage definition named "aix53J".

```
lsdef -t osimage -l -o aix53J
```

10. To list all node definitions that have a status value of "booting".

```
lsdef -t node -w status==booting
```

11. To list all the attributes of the group "service".

```
lsdef -l -t group -o service
```

12. To list all the attributes of the nodes that are members of the group "service".

```
lsdef -t node -l service
```

13. To get a listing of object definitions that includes information about what xCAT database tables are used to store the data.

```
lsdef -V -l -t node -o node01
```

14. To list the hidden nodes that can't be seen with other flags. The hidden nodes are FSP/BPAs.

```
lsdef -S
```

15. To list the nodes status and use xcoll to format the output.

```
lsdef -t node -i status -c | xcoll
```

16. To display the description for some specific attributes that could be used when defining an xCAT node.

```
lsdef -t node -h -i profile,pprofile
```

17. To display the nics configuration information for node cn1.

```
lsdef cn1 --nics
```

FILES

/opt/xcat/bin/lsdef

NOTES

This command is part of the xCAT software product.

SEE ALSO

mkdef(1)|mkdef.1, chdef(1)|chdef.1, rmdef(1)|rmdef.1, xcatstanzafile(5)|xcatstanzafile.5

lsflexnode.1

NAME

lsflexnode - Display the information of flexible node

SYNOPSIS

lsflexnode [-h | -****-help]

lsflexnode [-v | -****-version]

lsflexnode *noderange*

DESCRIPTION

IBM BladeCenter HX5 offers flexibility ideal that the blades can be combined together for scalability.

There are several concepts to support the HX5 multiple blades combination:

Complex: Multiple blades which combined by a scalability card is a complex.

Parition: A logic concept which containing part of the **Blade slot node** in a complex. Each partition can map to a system to install Operating System. Each partition could have 1HX5, 1HX5+1MD or 2HX5+2MD. (MD is the Memory Drawer)

Blade slot node: The physical blade which installed in the slot of a chassis. It can be a HX5 or MD.

A **Complex** will be created automatically when a multiple blades combination is installed. In this **Complex**, every blade belongs to it is working as a **Blade slot node**.

A **Partition** can be created base on the **Complex**, each **Partition** can have one or multiple **Blade slot node**.

The *noderange* in the **SYNOPSIS** can be a AMM node or a blade node.

OPTIONS

-h | --help

> Display the usage message.

-v | --version

> Display the version information.

ATTRIBUTES

The meaning of attributes which displayed by the **lsflexnode**. The word 'node' in this section means **Blade slot node**.

Complex

> The unique numeric identifier for a complex installed in the chassis.

Partition number

> The number of partitions currently defined for this complex.

Complex node number

> The number of nodes existing in this complex, regardless of their assignment to any given partition.

Partition

> The unique numeric identifier for a partition defined within a complex installed in the chassis.

Partition Mode

> The currently configured mode of this partition. It can be 'partition' or 'standalone'.

Partition node number

> The number of nodes currently defined for this partition.

Partition status

> The current power status of this partition when the partition has a valid partition configuration. It can be 'poweredoff', 'poweredon', 'resetting' or 'invalid'.

Node

> The unique numeric identifier for this node, unique within the partition. If this node does not belong to a partition, the slot number will be displayed.

Node state

> The physical power state of this node. It can be 'poweredoff', 'poweredon' or 'resetting'.

Node slot

> The base slot number where the node exists in the chassis.

Node resource

> A string providing a summary overview of the resources provided by this node. It includes the CPU number, CPU frequency and Memory size.

Node type

> The general categorization of the node. It can be 'processor', 'memory' or 'io'.

Node role

> Indicates if the node is assigned to a partition, and if so, provides an indication of whether the node is the primary node of the partition or not.

Flexnode state

> The state of a flexible node. It is the state of the partition which this node belongs to. If this node does NOT belong to a partition, the value should be 'invalid'.

> It can be 'poweredoff', 'poweredon', 'resetting' or 'invalid'.

Complex id

> The identifier of the complex this node belongs to.

Partition id

> The identifier of the partition this node belongs to.

EXAMPLES

1

> Display all the **Complex**, **Partition** and **Blade slot node** which managed by a AMM.

```
lsflexnode amm1
```

> The output:

```
amm1: Complex - 24068
amm1: ..Partition number - 1
amm1: ..Complex node number - 2
amm1: ..Partition = 1
```

```
amm1: ....Partition Mode - partition
amm1: ....Partition node number - 1
amm1: ....Partition status - poweredoff
amm1: ....Node - 0 (logic id)
amm1: ......Node state - poweredoff
amm1: ......Node slot - 14
amm1: ......Node type - processor
amm1: ......Node resource - 2 (1866 MHz) / 8 (2 GB)
amm1: ......Node role - secondary
amm1: ..Partition = unassigned
amm1: ....Node - 13 (logic id)
amm1: ......Node state - poweredoff
amm1: ......Node slot - 13
amm1: ......Node type - processor
amm1: ......Node resource - 2 (1866 MHz) / 8 (2 GB)
amm1: ......Node role - unassigned
```

2

Display a flexible node.

```
lsflexnode blade1
```

The output:

```
blade1: Flexnode state - poweredoff
blade1: Complex id - 24068
blade1: Partition id - 1
blade1: Slot14: Node state - poweredoff
blade1: Slot14: Node slot - 14
blade1: Slot14: Node type - processor
blade1: Slot14: Node resource - 2 (1866 MHz) / 8 (2 GB)
blade1: Slot14: Node role - secondary
```

FILES

/opt/xcat/bin/lsflexnode

SEE ALSO

mkflexnode(1)|mkflexnode.1, rmflexnode(1)|rmflexnode.1

lshwconn.1

NAME

lshwconn - Use this command to display the connection status for CEC and Frame nodes.

SYNOPSIS

lshwconn [**-h**| **--help**]

lshwconn [**-v**| **--version**]

PPC (with HMC) specific:

lshwconn [**-V**| **--verbose**] *noderange*

PPC (without HMC, using FSPAPI) specific:

lshwconn *noderange* **-T** *tooltype*

DESCRIPTION

This command is used to display the connection status for CEC and Frame node.

OPTIONS

-h|--help

> Display usage message.

-V|--verbose

> Verbose output.

-T

> The tooltype is used to communicate to the CEC/Frame. The value could be lpar or fnm. The tooltype
> value lpar is for xCAT and fnm is for CNM.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To display connection status for all CEC nodes in node group CEC:

   ```
   lshwconn cec
   ```

   Output is similar to:

   ```
   cec1: ipaddr=192.168.200.245,alt_ipaddr=unavailable,state=Connected
   cec2: Connection not found
   ```

2. To display connection status for Frame node frame1:

   ```
   lshwconn frame1
   ```

   Output is similar to:

```
frame1: side=a,ipaddr=192.168.200.247,alt_ipaddr=unavailable,state=Connected
frame1: side=b,ipaddr=192.168.200.248,alt_ipaddr=unavailable,state=Connected
```

3. To display connection status for all CEC nodes in node group CEC to hardware server, and using lpar tooltype:

```
lshwconn cec -T lpar
```

Output is similar to:

```
cec1: sp=primary,ipadd=40.3.7.1,alt_ipadd=unavailable,state=LINE UP
cec2: Connection not found
```

FILES

$XCATROOT/bin/lshwconn

(The XCATROOT environment variable is set when xCAT is installed. The default value is "/opt/xcat".)

NOTES

This command is part of the xCAT software product.

SEE ALSO

rmhwconn(1)|rmhwconn.1, mkhwconn(1)|mkhwconn.1

lskit.1

NAME

lskit - Lists information for one or more Kits.

SYNOPSIS

lskit [**-V** | **--verbose**] [**-F** | **--framework** *kitattr_names*] [**-x** | **--xml** | **--XML**] [**-K** | **--kitattr** *kitattr_names*] [**-R** | **--repoattr** *repoattr_names*] [**-C** | **--compattr** *compattr_names*] [kit_names]

lskit [**-?** | **-h** | **--help** | **-v** | **--version**]

lskit [**-F** | **--framework** *kit_path_name*]

DESCRIPTION

The **lskit** command is used to list information for one or more kits. A kit is a special kind of package that is used to install a software product on one or more nodes in an xCAT cluster.

Note: The xCAT support for Kits is only available for Linux operating systems.

The **lskit** command outputs the following info for each kit: the kit's basic info, the kit's repositories, and the kit's components. The command outputs the info in two formats: human-readable format (default), and XML format. Use the -x option to view the info in XML format.

Input to the command can specify any number or combination of the input options.

OPTIONS

-F|--framework *kit_path_name*

> Use this option to display the framework values of the specified Kit tarfile. This information is retrieved directly from the tarfile and can be done before the Kit has been defined in the xCAT database. This option cannot be combined with other options.

-K|--kitattr *kitattr_names*

> Where *kitattr_names* is a comma-delimited list of kit attribute names. The names correspond to attribute names in the **kit** table. The **lskit** command will only display the specified kit attributes.

-R|--repoattr *repoattr_names*

> Where *repoattr_names* is a comma-delimited list of kit repository attribute names. The names correspond to attribute names in the **kitrepo** table. The **lskit** command will only display the specified kit repository attributes.

-C|--compattr *compattr_names*

> where *compattr_names* is a comma-delimited list of kit component attribute names. The names correspond to attribute names in the **kitcomponent** table. The **lskit** command will only display the specified kit component attributes.

kit_names

> is a comma-delimited list of kit names. The **lskit** command will only display the kits matching these names.

-x|--xml|--XML

> Need XCATXMLTRACE=1 env when using -x|–xml|–XML, for example: XCATXMLTRACE=1 lskit -x testkit-1.0.0 Return the output with XML tags. The data is returned as:
>
> > **<data>**
> >
> > > **<kitinfo>** ...
> > >
> > > </kitinfo>
> >
> > </data> ... <data>
> >
> > > **<kitinfo>** ...
> > >
> > > </kitinfo>
> >
> > </data>

Each <kitinfo> tag contains info for one kit. The info inside <kitinfo> is structured as follows:
> The <kit> sub-tag contains the kit's basic info. The <kitrepo> sub-tags store info about the kit's repositories. The <kitcomponent> sub-tags store info about the kit's components.

The data inside <kitinfo> is returned as:

> **<kitinfo>**
>
> > **<kit>** ...

```
        </kit>
```

```
    <kitrepo>
      ...
    </kitrepo>
    ...

    <kitcomponent>
      ...
    </kitcomponent>
    ...
</kitinfo>
```

-V|--verbose

>   Display additional progress and error messages.

-v|--version

>   Command Version.

-?|-h|--help

>   Display usage message.

RETURN VALUE

0

>   The command completed successfully.

1

>   An error has occurred.

EXAMPLES

1. To list all kits, enter:

```
lskit
```

2. To list the kit "kit-test1-1.0-Linux", enter:

```
lskit kit-test1-1.0-Linux
```

3. To list the kit "kit-test1-1.0-Linux" for selected attributes, enter:

```
lskit -K basename,description -R kitreponame -C kitcompname kit-test1-1.0-Linux
```

4. To list the framework value of a Kit tarfile.

```
lskit -F /myhome/mykits/pperte-1.3.0.2-0-x86_64.tar.bz2

Extracting the kit.conf file from /myhome/mykits/pperte-1.3.0.2-0-x86_64.tar.bz2.␣
→Please wait.

    kitframework=2
    compatible_kitframeworks=0,1,2
```

5. To list kit "testkit-1.0-1" with XML tags, enter:

```
XCATXMLTRACE=1 lskit -x testkit-1.0-1
```

FILES

/opt/xcat/bin/lskit

SEE ALSO

lskitcomp(1)|lskitcomp.1, lskitdeployparam(1)|lskitdeployparam.1, addkit(1)|addkit.1, rmkit(1)|rmkit.1, addkit-comp(1)|addkitcomp.1, rmkitcomp(1)|rmkitcomp.1

lskitcomp.1

NAME

lskitcomp - Used to list information for one or more kit components.

SYNOPSIS

lskitcomp [-V | --verbose] [-x | --xml | --XML] [-C | --compattr *compattr_names*] **[-O | --osdistro** *os_distro*] **[-S | --serverrole** *server_role*] [kitcomp_names]

lskitcomp [-? | -h | --help | -v | --version]

DESCRIPTION

The **lskitcomp** command is used to list information for one or more kit components. A kit is made up of one or more kit components. Each kit component is a meta package used to install a software product component on one or more nodes in an xCAT cluster.

The **lskitcomp** command outputs the kit component info in two formats: human-readable format (default), and XML format. Use the -x option to view the info in XML format.

Input to the command can specify any number or combination of the input options.

Note: The xCAT support for Kits is only available for Linux operating systems.

OPTIONS

-C|--compattr *compattr_names*

> where *compattr_names* is a comma-delimited list of kit component attribute names. The names correspond to attribute names in the **kitcomponent** table. The **lskitcomp** command will only display the specified kit component attributes.

-O|--osdistro *os_distro*

> where *os_distro* is the name of an osdistro in **osdistro** table. The **lskitcomp** command will only display the kit components matching the specified osdistro.

-S|--serverrole *server_role*

> where *server_role* is the name of a server role. The typical server roles are: mgtnode, servicenode, computenode, loginnode, storagennode. The **lskitcomp** command will only display the kit components matching the specified server role.

kitcomp_names

> is a comma-delimited list of kit component names. The **lskitcomp** command will only display the kit components matching the specified names.

-x|--xml|--XML

> Need XCATXMLTRACE=1 env when using -x|–xml|–XML. Return the output with XML tags. The data is returned as:
>
> > **<data>**
> >
> > > **<kitinfo>** ...
> > >
> > > </kitinfo>
> >
> > </data> ... <data>
> >
> > > **<kitinfo>** ...
> > >
> > > </kitinfo>
> >
> > </data>

> **Each <kitinfo> tag contains info for a group of kit compoonents belonging to the same kit. The info inside <kitinfo> is stru**
> The <kit> sub-tag contains the kit's name. The <kitcomponent> sub-tags store info about the kit's components.

> **The data inside <kitinfo> is returned as:**
>
> > **<kitinfo>**
> >
> > > **<kit>** ...
> > >
> > > </kit>

```
    <kitcomponent>
      ...
    </kitcomponent>
    ...
</kitinfo>
```

-V|--verbose

> Display additional progress and error messages.

-v|--version

> Command Version.

-?|-h|--help

> Display usage message.

RETURN VALUE

0

> The command completed successfully.

1

An error has occurred.

EXAMPLES

1. To list all kit components, enter:

```
lskitcomp
```

2. To list the kit component "comp-server-1.0-1-rhels-6-x86_64", enter:

```
lskitcomp comp-server-1.0-1-rhels-6-x86_64
```

3. To list the kit component "comp-server-1.0-1-rhels-6-x86_64" for selected kit component attributes, enter:

```
lskitcomp -C kitcompname,desc comp-server-1.0-1-rhels-6-x86_64
```

4. To list kit components compatible with "rhels-6.2-x86_64" osdistro, enter:

```
lskitcomp -O rhels-6.2-x86_64
```

5. To list kit components compatible with "rhels-6.2-x86_64" osdistro and "computenode" server role, enter:

```
lskitcomp -O rhels-6.2-x86_64 -S computenode
```

6. To list the kit component "testkit-compute-1.0-1-ubuntu-14.04-ppc64el" with XML tags, enter:

```
XCATXMLTRACE=1 lskitcomp -x testkit-compute-1.0-1-ubuntu-14.04-ppc64el
```

FILES

/opt/xcat/bin/lskitcomp

SEE ALSO

lskit(1)|lskit.1, lskitdeployparam(1)|lskitdeployparam.1, addkit(1)|addkit.1, rmkit(1)|rmkit.1, addkit-comp(1)|addkitcomp.1, rmkitcomp(1)|rmkitcomp.1

lskitdeployparam.1

NAME

lskitdeployparam - Lists the deployment parameters for one or more Kits or Kit components

SYNOPSIS

lskitdeployparam [-V | --verbose] **[-x | --xml | --XML] [-k | --kitname** *kit_names*] [-c | --compname *comp_names*]

lskitdeployparam [-? | -h | --help | -v | --version]

DESCRIPTION

The **lskitdeployparam** command is used to list the kit deployment parameters for one or more kits, or one or more kit components. Kit deployment parameters are used to customize the installation or upgrade of kit components.

The **lskitdeployparam** command outputs the kit component information in two formats: human-readable format (default), and XML format. Use the -x option to view the information in XML format.

Input to the command can specify any combination of the input options.

Note: The xCAT support for Kits is only available for Linux operating systems.

OPTIONS

-k|--kitname *kit_names*

Where *kit_names* is a comma-delimited list of kit names. The **lskitdeployparam** command will only display the deployment parameters for the kits with the matching names.

-c|--compname *comp_names*

Where *comp_names* is a comma-delimited list of kit component names. The **lskitdeployparam** command will only display the deployment parameters for the kit components with the matching names.

-x|--xml|--XML

Return the output with XML tags. The data is returned as:

<data>

<kitdeployparam> <name>KIT_KIT1_PARAM1</name> <value>value11</value>

</kitdeployparam>

</data> <data>

<kitdeployparam> <name>KIT_KIT1_PARAM2</name> <value>value12</value>

</kitdeployparam>

</data> ...

-V|--verbose

Display additional progress and error messages.

-v|--version

Command Version.

-?|-h|--help

Display usage message.

RETURN VALUE

0

The command completed successfully.

1

An error has occurred.

EXAMPLES

1. To list kit deployment parameters for kit "kit-test1-1.0-Linux", enter:

   ```
   lskitdeployparam -k kit-test1-1.0-Linux
   ```

2. To list kit deployment parameters for kit component "comp-server-1.0-1-rhels-6-x86_64", enter:

   ```
   lskitdeployparam -c comp-server-1.0-1-rhels-6-x86_64
   ```

FILES

/opt/xcat/bin/lskitdeployparam

SEE ALSO

lskit(1)|lskit.1, lskitcomp(1)|lskitcomp.1, addkit(1)|addkit.1, rmkit(1)|rmkit.1, addkitcomp(1)|addkitcomp.1, rmkit-comp(1)|rmkitcomp.1

lskmodules.1

NAME

lskmodules - list kernel driver modules in rpms or driver disk image files

SYNOPSIS

lskmodules [-V | --verbose] **[-i | --osimage** *osimage_names*] **[-c | --kitcomponent** *kitcomp_names*] **[-o | --osdistro** *osdistro_names*] **[-u | --osdistropudate** *osdistroupdate_names*] **[-x | --xml | --XML]**

lskmodules [-? | -h | --help | -v | --version]

DESCRIPTION

The **lskmodules** command finds the kernel driver module files (*.ko) in the specified input locations, runs the modinfo command against each file, and returns the driver name and description. If -x is specified, the output is returned with XML tags.

Input to the command can specify any number or combination of the input options.

OPTIONS

-i|--osimage *osimage_names*

> where *osimage_names* is a comma-delimited list of xCAT database osimage object names. For each *osimage_name*, lskmodules will use the entries in osimage.driverupdatesrc for the rpms and driver disk image files to search.

-c|--kitcomponent *kitcomponent_names*

where *kitcomponent_names* is a comma-delimited list of xCAT database kitcomponent object names. For each *kitcomponent_name*, lskmodules will use the entries in kitcomponent.driverpacks for the rpm list and the repodir of the kitcomponent.kitreponame for the location of the rpm files to search.

-o|--osdistro *osdistro_names*

where *osdistro_names* is a comma-delimited list of xCAT database osdistro object names. For each *osdistro_name*, lskmodules will search each <osdistro.dirpaths>/Packages/kernel-<kernelversion>.rpm file.

-u|--osdistroupdate *osdistroupdate_names*

where *osdistroupdate_names* is a comma-delimited list of xCAT database osdistroupdate table entries. For each *osdistroupdate_name*, lskmodules will search the <osdistroupdate.dirpath>/kernel-<kernelversion>.rpm file.

-x|--xml|--XML

> **Return the output with XML tags. The data is returned as:**

>> **<module>** <name> xxx.ko </name> <description> this is module xxx </description>

>> </module>

This option is intended for use by other programs. The XML will not be displayed. To view the returned XML, set the XCATSHOWXML=yes environment variable before running this command.

-V|--verbose

Display additional progress and error messages.

-v|--version

Command Version.

-?|-h|--help

Display usage message.

RETURN VALUE

0

The command completed successfully.

1

An error has occurred.

EXAMPLES

1. To list the kernel modules included in the driverpacks shipped with kitcomponent kit1_comp1-x86_64, enter:

```
lskmodules -c kit1_comp1-x86_64
```

FILES

SEE ALSO

lslite.1

NAME

lslite - Display a summary of the statelite information.

SYNOPSIS

lslite [-h | -****-help]

lslite [-V | -****-verbose] [-i imagename] | [noderange]

DESCRIPTION

The **lslite** command displays a summary of the statelite information that has been defined for a noderange or an image.

OPTIONS

-h|--help

> Display usage message.

-V|--verbose

> Verbose mode.

-i imagename

> The name of an existing xCAT osimage definition.

noderange

> A set of comma delimited node names and/or group names. See the "noderange" man page for details on additional supported formats.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To list the statelite information for an xCAT node named "node01".

   lslite node01

   Output is similar to:

   >>>Node: node01

   Osimage: 61img

   Persistent directory (statelite table): xcatmn1:/statelite

   Litefiles (litefile table): tmpfs,rw /etc/adjtime tmpfs,rw /etc/lvm/.cache tmpfs,rw /etc/mtab ....

Litetree path (litetree table): 1,MN:/etc 2,server1:/etc

2. To list the statelite information for an xCAT osimage named "osimage01".

lslite -i osimage01

Output is similar to:

```
tmpfs,rw       /etc/adjtime
tmpfs,rw       /etc/lvm/.cache
tmpfs,rw       /etc/mtab
........
```

FILES

/opt/xcat/bin/lslite

SEE ALSO

noderange(3)|noderange.3, tabdump(8)|tabdump.8

lsslp.1

NAME

lsslp - Discovers selected networked services information within the same subnet.

SYNOPSIS

lsslp [-h| –help]

lsslp [-v| –version]

lsslp [noderange] [-V] [-i ip[,ip..]][-w][-r|-x|-z][-n][-s CEC|FRAME|MM|IVM|RSA|HMC|CMM|IMM2|FSP][-t tries][-I][-C counts][-T timeout][–vpdtable]

DESCRIPTION

The lsslp command discovers selected service types using the -s flag. All service types are returned if the -s flag is not specified. If a specific IP address is not specified using the -i flag, the request is sent out all available network adapters. The optional -r, -x, -z and –vpdtable flags format the output. If you can't receive all the hardware, please use -T to increase the waiting time.

NOTE: SLP broadcast requests will propagate only within the subnet of the network adapter broadcast IPs specified by the -i flag.

OPTIONS

noderange The nodes which the user want to discover. If the user specify the noderange, lsslp will just return the nodes in the node range. Which means it will help to add the new nodes to the xCAT database without modifying the existed definitions. But the nodes' name specified in noderange should be defined in database in advance.

The specified nodes' type can be frame/cec/hmc/fsp/bpa. If the it is frame or cec, lsslp will list the bpa or fsp nodes within the nodes(bap for frame, fsp for cec). Please do not use noderange with the flag -s.

-i IP(s) the command will send out (defaults to all available adapters).

-h Display usage message.

-n Only display and write the newly discovered hardwares.

-u Do unicast to a specified IP range. Must be used with -s and -**-range.** The -u flag is not supported on AIX.

--range Specify one or more IP ranges. Must be use in unicast mode. It accepts multiple formats. For example, 192.168.1.1/24, 40-41.1-2.3-4.1-100. If the range is huge, for example, 192.168.1.1/8, lsslp may take a very long time for node scan. So the range should be exactly specified.

-r Display Raw SLP response.

-C The number of the expected responses specified by the user. When using this flag, lsslp will not return until the it has found all the nodes or time out. The default max time is 3 secondes. The user can use -T flag the specify the time they want to use. A short time will limite the time costing, while a long time will help to find all the nodes.

-T The number in seconds to limite the time costing of lsslp.

-s Service type interested in discovering.

-t Number or service-request attempts.

--vpdtable Output the SLP response in vpdtable formatting. Easy for writting data to vpd table.

-v Command Version.

-V Verbose output.

-w Writes output to xCAT database.

-x XML format.

-z Stanza formated output.

-I Give the warning message for the nodes in database which have no SLP responses. Please note that this flag noly can be used after the database migration finished successfully.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To list all discovered HMC service types in tabular format, enter:

```
lsslp -s HMC
```

Output is similar to:

```
device type-model serial-number ip-addresses   hostname
HMC     7310CR2    103F55A       1.1.1.115      hmc01
HMC     7310CR2    105369A       3.3.3.103      hmc02
HMC     7310CR3    KPHHK24       3.3.3.154      hmc03
```

2. list all discovered FSP service types in raw response format on subnet 30.0.0.255, enter:

```
lsslp -i 3.0.0.255 -s CEC -r
```

Output is similar to:

```
(type=cec-service-processor),(serial-number=10A3AEB),(machinetype-model=9117-570),
→(fru-serial-number=YL11C5338102),(hostname=),(frame-number=0),(cage-number=0),(ip-
→address=3.0.0.94,1.1.1.147),(web-url=https://3.0.0.94:473 ), (slot=1),(bpc-
→machinetype-model=0),(bpc-serial-number=0),(Image=fips240/b0630a_0623.240)
(type=cec-service-processor),(serial-number=10A3E2B),(machinetype-model=9117-570),
→(fru-serial- number=YL11C5338250),(hostname=),(frame-number=0),(cage-number=0),(ip-
→address=3.0.0.95,1.1.1.147), (web-url=https://3.0.0.95:473 ),(slot=1),(bpc-
→machinetype-model=0),(bpc-serial-number=0),(Image=fips240/b0630a_0623.240)
```

3. To list all discovered MM service types in XML format and write the output to the xCAT database, enter:

```
lsslp -s MM -x -w
```

Output is similar to:

```
<Node>
  <groups>mm,all</groups>
  <id>00:14:5E:E0:CB:1E</id>
  <mgt>blade</mgt>
  <mtm>029310C</mtm>
  <node>Server-029310C-SN100485A-A</node>
  <nodetype>mm</nodetype>
  <otherinterfaces>9.114.47.229</otherinterfaces>
  <serial>100485A</serial>
</Node>
```

4. To list all discovered service types in stanza format and write the output to the xCAT database, enter:

```
lsslp -z -w
```

Output is similar to:

c76v1hmc02: objtype=node hcp=c76v1hmc02 nodetype=hmc mtm=7315CR2 serial=10407DA ip=192.168.200.125 groups=hmc,all mgt=hmc mac=00:1a:64:fb:7d:50 hidden=0

192.168.200.244: objtype=node hcp=192.168.200.244 nodetype=fsp mtm=9125-F2A serial=0262662 side=A-0 otherinterfaces=192.168.200.244 groups=fsp,all mgt=fsp id=4 parent=Server-9125-F2A-SN0262662 mac=00:1a:64:fa:01:fe hidden=1

Server-8205-E6B-SN1074CDP: objtype=node hcp=Server-8205-E6B-SN1074CDP nodetype=cec mtm=8205-E6B serial=1074CDP groups=cec,all mgt=fsp id=0 hidden=0

192.168.200.33: objtype=node hcp=192.168.200.33 nodetype=bpa mtm=9458-100 serial=99201WM side=B-0 otherinterfaces=192.168.200.33 groups=bpa,all mgt=bpa id=0 mac=00:09:6b:ad:19:90 hidden=1

Server-9125-F2A-SN0262652: objtype=node hcp=Server-9125-F2A-SN0262652 nodetype=frame mtm=9125-F2A serial=0262652 groups=frame,all mgt=fsp id=5 hidden=0

5. To list all discovered service types in stanza format and display the IP address, enter:

```
lsslp -w
```

Output is similar to:

```
mm01:
   objtype=node
   nodetype=fsp
   mtm=8233-E8B
   serial=1000ECP
   side=A-0
   groups=fsp,all
   mgt=fsp
   id=0
   mac=00:14:5E:F0:5C:FD
   otherinterfaces=50.0.0.5

bpa01:
   objtype=node
   nodetype=bpa
   mtm=9A01-100
   serial=0P1N746
   side=A-1
   groups=bpa,all
   mgt=bpa
   id=0
   mac=00:1A:64:54:8C:A5
   otherinterfaces=50.0.0.1
```

6. To list all the CECs, enter:

```
lsslp -s CEC
```

device type-model serial-number side ip-addresses hostname FSP 9117-MMB 105EBEP A-1 20.0.0.138 20.0.0.138 FSP 9117-MMB 105EBEP B-1 20.0.0.139 20.0.0.139 CEC 9117-MMB 105EBEP Server-9117-MMB-SN105EBEP

7. To list all the nodes defined in database which have no SLP response.

```
lsslp -I
```

Output is similar to:

These nodes defined in database but can't be discovered: f17c00bpcb_b,f17c01bpcb_a,f17c01bpcb_b,f17c02bpcb_a,

device type-model serial-number side ip-addresses hostname bpa 9458-100 BPCF017 A-0 40.17.0.1 f17c00bpca_a bpa 9458-100 BPCF017 B-0 40.17.0.2 f17c00bpcb_a

8. To find the nodes within the user specified. Please make sure the noderange input have been defined in xCAT database.

```
   lsslp CEC1-CEC3
or lsslp CEC1,CEC2,CEC3

  device    type-model    serial-number    side    ip-addresses       hostname
  FSP       9A01-100      0P1P336          A-0     192.168.200.34    192.168.200.34
  FSP       9A01-100      0P1P336          B-0     192.168.200.35    192.168.200.35
  FSP       9A01-100      0P1P336          A-1     50.0.0.27          50.0.0.27
  FSP       9A01-100      0P1P336          B-1     50.0.0.28          50.0.0.28
  CEC       9A01-100      0P1P336                                     CEC1
  FSP       8233-E8B      1040C7P          A-0     192.168.200.36    192.168.200.36
  FSP       8233-E8B      1040C7P          B-0     192.168.200.37    192.168.200.37
  FSP       8233-E8B      1040C7P          A-1     50.0.0.29          50.0.0.29
  FSP       8233-E8B      1040C7P          B-1     50.0.0.30          50.0.0.30
  CEC       8233-E8B      1040C7P                                     CEC2
```

```
FSP        8205-E6B       1000ECP         A-0    192.168.200.38  192.168.200.38
FSP        8205-E6B       1000ECP         B-0    192.168.200.39  192.168.200.39
FSP        8205-E6B       1000ECP         A-1    50.0.0.31       50.0.0.27
FSP        8205-E6B       1000ECP         B-1    50.0.0.32       50.0.0.28
CEC        8205-E6B       1000ECP                                CEC3
```

9. To list all discovered CMM in stanza format, enter: lsslp -s CMM -m -z

e114ngmm1: objtype=node mpa=e114ngmm1 nodetype=cmm mtm=98939AX serial=102537A groups=cmm,all
mgt=blade hidden=0 otherinterfaces=70.0.0.30 hwtype=cmm

10. To use lsslp unicast, enter: lsslp -u -s CEC –range 40-41.1-2.1-2.1-2

FILES

/opt/xcat/bin/lsslp

SEE ALSO

rscan(1)|rscan.1

lstree.1

NAME

lstree - Display the tree of service node hierarchy, hardware hierarchy, or VM hierarchy.

SYNOPSIS

lstree [-h | -****-help]

lstree [-s | -**-servicenode] [-H | --hardwaremgmt] [-v | -**-virtualmachine] [noderange]

DESCRIPTION

The **lstree** command can display the tree of service node hierarchy for the xCAT nodes which have service node
defined or which are service nodes, display the tree of hardware hierarchy only for the physical objects, display the
tree of VM hierarchy for the xCAT nodes which are virtual machines or which are the hosts of virtual machines. If a
noderange is specified, only show the part of the hierarchy that involves those nodes. For ZVM, we only support to
displ y VM hierarchy. By default, lstree will show both the hardware hierarchy and the VM hierarchy for all the nodes.

OPTIONS

-h|--help

   Display usage message.

-s|-- servicenode

   Show the tree of service node hierarchy.

-H|--hardwaremgmt

> Show the tree of hardware hierarchy.

--v|--virtualmachine

> Show the tree of VM hierarchy.

noderange

> A set of comma delimited node names and/or group names. See the "noderange" man page for details on additional supported formats.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To display the tree of service node hierarchy for all the nodes.

   lstree -s

   Output is similar to:

   Service Node: mysn01 |__mycn01 |__mycn02 |__mycn03

   Service Node: mysn02 |__mycn11 |__mycn12 |__mycn13 ......

2. To display the tree of service node hierarchy for service node "mysn01".

   lstree -s mysn01

   Output is similar to:

   Service Node: mysn01 |__mycn01 |__mycn02 |__mycn03

3. To display the tree of hardware hierarchy for all the nodes.

   lstree -H

   Output is similar to:

   HMC: myhmc01

   > **|__Frame: myframe01** |__CEC: mycec01 |__CEC: mycec02 ......

   Service Focal Point: myhmc02

   > **|__Frame: myframe01** |__CEC: mycec01 |__CEC: mycec02 |__CEC: mycec03 ......

   Management Module: mymm01 |__Blade 1: js22n01 |__Blade 2: js22n02 |__Blade 3: js22n03 ......

   BMC: 192.168.0.1 |__Server: x3650n01

4. To display the tree of hardware hierarchy for HMC "myhmc01".

   lstree -H myhmc01

   Output is similar to:

   HMC: myhmc01

   > **|__Frame: myframe01** |__CEC: mycec01 |__CEC: mycec02 ......

---

5. To display the tree of VM hierarchy for all the nodes.

   lstree -v

   Output is similar to:

   Server: hs22n01 **|__** hs22vm1

   Server: x3650n01 **|__** x3650n01kvm1 **|__** x3650n01kvm2

6. To display the tree of VM hierarchy for the node "x3650n01".

   lstree -v x3650n01

   Output is similar to:

   Server: x3650n01 **|__** x3650n01kvm1 **|__** x3650n01kvm2

7. To display both the hardware tree and VM tree for all nodes.

   lstree

   Output is similar to:

   HMC: myhmc01

   > **|__Frame: myframe01**

   > > **|__CEC: mycec01** **|__**LPAR 1: node01 **|__**LPAR 2: node02 **|__**LPAR 3: node03 ......

   > > **|__CEC: mycec02** **|__**LPAR 1: node11 **|__**LPAR 2: node12 **|__**LPAR 3: node13 ......

   Service Focal Point: myhmc02

   > **|__Frame: myframe01**

   > > **|__CEC: mycec01** **|__**LPAR 1: node01 **|__**LPAR 2: node02 **|__**LPAR 3: node03 ......

   > **|__Frame: myframe02**

   > > **|__CEC: mycec02** **|__**LPAR 1: node21 **|__**LPAR 2: node22 **|__**LPAR 3: node23 ......

   Management Module: mymm01

   > **|__Blade 1: hs22n01** **|__**hs22n01vm1 **|__**hs22n01vm2

   > **|__Blade 2: hs22n02** **|__**hs22n02vm1 **|__**hs22n02vm2

   BMC: 192.168.0.1

   > **|__Server: x3650n01** **|__** x3650n01kvm1 **|__** x3650n01kvm2

FILES

/opt/xcat/bin/lstree

SEE ALSO

noderange(3)|noderange.3, tabdump(8)|tabdump.8

lsve.1

NAME

lsve - Lists detail attributes for a virtual environment.

SYNOPSIS

lsve [**-t** type] [**-m** manager] [**-o** object]

DESCRIPTION

The **lsve** command can be used to list a virtual environment for 'Data Center', 'Cluster', 'Storage Domain', 'Network' and 'Template' objects.

The mandatory parameter **-m manager** is used to specify the address of the manager of virtual environment. xCAT needs it to access the RHEV manager.

The mandatory parameter **-t type** is used to specify the type of the target object.

Basically, **lsve** command supports three types of object: **dc**, **cl**, **sd**, **nw**and **tpl**.

The parameter **-o object** is used to specify which object to list. If no **-o** is specified, all the objects with the **-t** type will be displayed.

OPTIONS

-h Display usage message.

-m Specify the manager of the virtual environment.

> For RHEV, the FQDN (Fully Qualified Domain Name) of the rhev manager have to be specified.

-o The target object to display.

-t Specify the **type** of the target object.

> **Supported types:**
>
> > **dc - Data Center (For type of 'dc', all the elements belongs to the data** center will be listed.)
> >
> > **cl** - Cluster **sd** - Storage Domain (To get the status of Storage Doamin, show it from
> >
> > > *data center* it attached to.
> >
> > **nw** - Network **tpl** - Template

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To list the data center 'Default', enter:

```
lsve -t B<dc> -m <FQDN of rhev manager> -o Default
```

Output is similar to:

```
datacenters: [Default]
description: The default Data Center
state: up
storageformat: v1
storagetype: nfs
  clusters: [Default]
    cpu: Intel Westmere Family
    description: The default server cluster
    memory_hugepage: true
    memory_overcommit: 100
  storagedomains: [image]
    available: 55834574848
    committed: 13958643712
    ismaster: true
    status: active
    storage_add: <Address of storage domain>
    storage_format: v1
    storage_path: /vfsimg
    storage_type: nfs
    type: data
    used: 9663676416
  networks: [rhevm2]
    description:
    state: operational
    stp: false
  networks: [rhevm]
    description: Management Network
    state: operational
    stp: false
  templates: [Blank]
    bootorder: hd
    cpucore: 1
    cpusocket: 1
    creation_time: 2008-04-01T00:00:00.000-04:00
    display: spice
    memory: 536870912
    state: ok
    stateless: false
    type: desktop
```

2. To list the cluster 'Default', enter:

```
lsve -t B<cl> -m <FQDN of rhev manager> -o Default
```

Output is similar to:

```
cpu: Intel Westmere Family
description: The default server cluster
memory_hugepage: true
memory_overcommit: 10
```

3. To list the Storage Domain 'image', enter:

```
lsve -t B<sd> -m <FQDN of rhev manager> -o image
```

Output is similar to:

> **storagedomains: [image]** available: 55834574848 committed: 13958643712 ismaster: true status: storage_add: <Address of storage domain> storage_format: v1 storage_path: /vfsimg storage_type: nfs type: data used: 9663676416

4. To list the network 'rhevm', enter:

```
lsve -t B<nw> -m <FQDN of rhev manager> -o rhevm
```

Output is similar to:

```
networks: [rhevm]
  description: Management Network
  state: operational
  stp: false
```

5. To list the template 'tpl01', enter:

```
lsve -t tpl -m <FQDN of rhev manager> -o tpl01
```

Output is similar to:

```
templates: [tpl01]
  bootorder: network
  cpucore: 2
  cpusocket: 2
  creation_time: 2012-08-22T23:52:35.953-04:00
  display: vnc
  memory: 1999634432
  state: ok
  stateless: false
  type: server
```

FILES

/opt/xcat/bin/lsve

SEE ALSO

cfgve(1)|cfgve.1

lsvlan.1

NAME

lsvlan - It lists the existing vlans for the cluster.

SYNOPSIS

lsvlan

lsvlan [*vlanid*]

lsvlan [**-h** | **--help**]

lsvlan [**-v** | **--version**]

DESCRIPTION

The **lsvlan** command lists all the vlans for the cluster. If *vlanid* is specifined it will list more details about this vlan including the nodes in the vlan.

Parameters

vlanid is a unique vlan number. If it is omitted, all vlans will be listed.

OPTIONS

-h|--help Display usage message.

-v|--version Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To list all the vlans in the cluster

   ```
   lsvlan
   ```

   Output is similar to:

   > **vlan 3:** subnet 10.3.0.0 netmask 255.255.0.0

   ```
   vlan 4:
       subnet 10.4.0.0
       netmask 255.255.0.0
   ```

2. TO list the details for vlan3

   ```
   lsvlan 3
   ```

   Output is similar to:

   > **vlan 3** subnet 10.3.0.0 netmask 255.255.0.0

```
hostname      ip address      node            vm host
v3n1          10.3.0.1        c68m4hsp06
v3n2          10.3.0.2        x3455n01
v3n3          10.3.0.3        x3650n01
v3n4          10.3.0.4        x3650n01kvm1    x3650n01
v3n5          10.3.0.5        x3650n01kvm2    x3650n01
```

FILES

/opt/xcat/bin/lsvlan

SEE ALSO

mkvlan(1)|mkvlan.1, rmvlan(1)|rmvlan.1, chvlan(1)|chvlan.1

lsvm.1

NAME

lsvm - Lists partition profile information for HMC-, DFM-, IVM-, KVM-, Vmware- and zVM-managed nodes. For Power 775, it lists the LPARs' I/O slots information and CEC configuration.

SYNOPSIS

lsvm [**-h**| **--help**]

lsvm [**-v**| **--version**]

lsvm [**-V**| **--verbose**] *noderange*

lsvm [**-a**| **--all**] *noderange*

For PPC (using Direct FSP Management):

lsvm [**-l**| **--long**] **--p775** *noderange*

lsvm *noderange*

For zVM:

lsvm *noderange*

DESCRIPTION

The lsvm command lists all partition profiles defined for the partitions specified in noderange. If noderange is a CEC, all the partitions associated with that CEC are displayed.

For PPC (using Direct FSP Management):

For Power 775(use option *–p775* to specify), lsvm lists all partition I/O slots information for the partitions specified in noderange. If noderange is a CEC, it gets the CEC's pump mode value, octant's memory interleaving value, the all the octants configure value, and all the I/O slots information.

For DFM-managed (short for Direct FSP Management mode) normal power machine, lsvm lists the processor, memory, physical I/O slots, hugepage and BSR info for the specified partitions or CEC.

The pump mode value has the valid options: 1 - Node Pump Mode 2 - Chip Pump Mode

The Memory Interleaving Mode has 3 valid options: 0 - not Applicable 1 - interleaved 2 - non-interleaved

More information about this part, refer to the section Using the *vm commands to define partitions in xCAT DFM in the doc be
XCAT_Power_775_Hardware_Management

For KVM and Vmware

The virtual machines that defined in the hypervisor *noderange* will be displayed. *noderange* only can be hypervisor. The type of the hypervisor should be set: hypervisor.type before running the lsvm.

Note: Only the virtual machine which is in power on state can be listed by lsvm command.

For zVM:

Show the directory entry for a given virtual machine.

OPTIONS

-h

Display usage message.

-v

Command Version.

-V

Verbose output.

-a

List all the profiles for one partition

--p775

Specify the operation is for Power 775 machines.

-l

Show lparnames for lpars. It shall work with option **--p775**.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To list all partition profiles defined for HMC-managed partition lpar3, enter:

```
lsvm lpar3
```

Output is similar to:

```
lpar3: name=lpar3,lpar_name=lpar3,lpar_id=4,lpar_env=aixlinux,all_resources=0,min_
→mem=512, desired_mem=2048, max_mem=3072,min_num_huge_pages=0,desired_num_huge_
→pages=0,max_num_huge_pages=0,proc_mode=shared, min_proc_units=0.5,desired_proc_
→units=0.5,max_proc_units=0.5,min_procs=1,desired_procs=1,max_procs=1, sharing_
→mode=uncap,uncap_weight=128,shared_proc_pool_id=0,shared_proc_pool_name=DefaultPool,
→io_slots=none, lpar_io_pool_ids=none,max_virtual_slots=10, "virtual_serial_
→adapters=1/server/1/any//any/1,0/server/1/any//any/1", virtual_scsi_adapters=2/
→client/1/p6vios/4/1,virtual_eth_adapters=3/0/1//0/1,hca_adapters=none,boot_
→mode=norm,conn_monitoring=0,auto_start=0,power_ctrl_lpar_ids=none,work_group_
→id=none,redundant_err_path_reporting=0, bsr_arrays=0,lhea_logical_ports=none,lhea_
→capabilities=none,lpar_proc_compat_mode=default,electronic_err_reporting=null
```

2.To list all IVM-managed partitions associated with CEC cec01, enter:

```
lsvm cec01
```

g Output is similar to:

```
cec01: name=10-B7D1G,lpar_name=10-B7D1G,lpar_id=1,os_type=vioserver,all_resources=0,
→min_mem=512, desired_mem=2048,max_mem=2048,proc_mode=shared,min_proc_units=0.10,
→desired_proc_units=0.40, max_proc_units=4.00,min_procs=1,desired_procs=4,max_
→procs=4,sharing_mode=uncap,uncap_weight=128, "io_slots=21010002/none/0,21010003/
→none/0,21010004/none/0,21020003/none/0,21020004/none/0,21030003/none/0,21030004/
→none/0,21040003/none/0,21040004/none/0",lpar_io_pool_ids=none,max_virtual_slots=48,
→"virtual_serial_adapters=0/server/1/any//any/1,1/server/1/any//any/1,10/client/0/2/
→lp2/0/0,12/client/0/3/lp3/0/0,14/client/0/4/lp4/0/0","virtual_scsi_adapters=11/
→server/2/lp2/2/0,13/server/3/lp3/2/0,15/server/4/lp4/2/0","virtual_eth_adapters=3/0/
→1//1/0,4/0/2//1/0,5/0/3//1/0,6/0/4//1/0",boot_mode=norm,conn_monitoring=0,auto_
→start=0,power_ctrl_lpar_ids=none
  name=lp2,lpar_name=lp2,lpar_id=2,os_type=aixlinux,all_resources=0,min_mem=128,
→desired_mem=1024,max_mem=1024,proc_mode=shared,min_proc_units=0.10,desired_proc_
→units=0.10,max_proc_units=4.00,min_procs=1,desired_procs=1,max_procs=4,sharing_
→mode=uncap,uncap_weight=128,io_slots=none,lpar_io_pool_ids=none,max_virtual_slots=6,
→ "virtual_serial_adapters=0/server/1/any//any/1,1/server/1/any//any/1",virtual_scsi_
→adapters=2/client/1/10-7D1G/11/1,virtual_eth_adapters=4/0/1//0/0,boot_mode=norm,
→conn_monitoring=0,auto_start=0,power_ctrl_lpar_ids=none
  name=lp3,lpar_name=lp3,lpar_id=3,os_type=aixlinux,all_resources=0,min_mem=128,
→desired_mem=128,max_mem=128,proc_mode=shared,min_proc_units=0.10,desired_proc_
→units=0.10,max_proc_units=4.00,min_procs=1,desired_procs=1,max_procs=4,sharing_
→mode=uncap,uncap_weight=128,io_slots=none,lpar_io_pool_ids=none,max_virtual_slots=6,
→ "virtual_serial_adapters=0/server/1/any//any/1,1/server/1/any//any/1",virtual_scsi_
→adapters=2/client/1/10-B7D1G/13/1,virtual_eth_adapters=4/0/1//0/0,boot_mode=of,conn_
→monitoring=0,auto_start=1, power_ctrl_lpar_ids=none
```

3. For Power 775, to list the I/O slot information of lpar1, enter:

```
lsvm lpar1 --p775
```

Output is similar to:

```
1: 514/U78A9.001.0123456-P1-C17/0x21010202/2/1
1: 513/U78A9.001.0123456-P1-C15/0x21010201/2/1
1: 512/U78A9.001.0123456-P1-C16/0x21010200/2/1
```

To list the lparname of lpars, enter:

```
lsvm lpar1 -l --p775
```

Output is similar to: lpar1: 1: 514/U78A9.001.0123456-P1-C17/0x21010202/2/1 lpar1: 1: 513/U78A9.001.0123456-P1-C15/0x21010201/2/1 lpar1: 1: 512/U78A9.001.0123456-P1-C16/0x21010200/2/1

4. For Power 775, to list the I/O slot information and octant configuration of cec1, enter:

```
lsvm cec1 --p775
```

Output is similar to:

```
1: 514/U78A9.001.0123456-P1-C17/0x21010202/2/1
1: 513/U78A9.001.0123456-P1-C15/0x21010201/2/1
1: 512/U78A9.001.0123456-P1-C16/0x21010200/2/1
13: 537/U78A9.001.0123456-P1-C9/0x21010219/2/13
13: 536/U78A9.001.0123456-P1-C10/0x21010218/2/13
17: 545/U78A9.001.0123456-P1-C7/0x21010221/2/17
17: 544/U78A9.001.0123456-P1-C8/0x21010220/2/17
21: 553/U78A9.001.0123456-P1-C5/0x21010229/2/21
21: 552/U78A9.001.0123456-P1-C6/0x21010228/2/21
25: 569/U78A9.001.0123456-P1-C1/0x21010239/2/25
25: 561/U78A9.001.0123456-P1-C3/0x21010231/2/25
25: 560/U78A9.001.0123456-P1-C4/0x21010230/2/25
29: 568/U78A9.001.0123456-P1-C2/0x21010238/2/29
5: 521/U78A9.001.0123456-P1-C13/0x21010209/2/5
5: 520/U78A9.001.0123456-P1-C14/0x21010208/2/5
9: 529/U78A9.001.0123456-P1-C11/0x21010211/2/9
9: 528/U78A9.001.0123456-P1-C12/0x21010210/2/9
cec1: PendingPumpMode=1,CurrentPumpMode=1,OctantCount=8:
OctantID=0,PendingOctCfg=5,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=1,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=2,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=3,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=4,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=5,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=6,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=7,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
```

To list the lparname of lpars, enter:

```
lsvm cec1 -l --p775
```

Output is similar to:

```
lpar1: 1: 514/U78A9.001.0123456-P1-C17/0x21010202/2/1: 32: 0/3/3
lpar1: 1: 513/U78A9.001.0123456-P1-C15/0x21010201/2/1: 32: 0/3/3
lpar1: 1: 512/U78A9.001.0123456-P1-C16/0x21010200/2/1: 32: 0/3/3
lpar13: 13: 537/U78A9.001.0123456-P1-C9/0x21010219/2/13: 32: 0/3/3
lpar13: 13: 536/U78A9.001.0123456-P1-C10/0x21010218/2/13: 32: 0/3/3
lpar17: 17: 545/U78A9.001.0123456-P1-C7/0x21010221/2/17: 32: 0/0/0
lpar17: 17: 544/U78A9.001.0123456-P1-C8/0x21010220/2/17: 32: 0/0/0
lpar21: 21: 553/U78A9.001.0123456-P1-C5/0x21010229/2/21: 32: 0/0/0
lpar21: 21: 552/U78A9.001.0123456-P1-C6/0x21010228/2/21: 32: 0/0/0
lpar24: 25: 569/U78A9.001.0123456-P1-C1/0x21010239/2/25: 32: 0/0/0
lpar25: 25: 561/U78A9.001.0123456-P1-C3/0x21010231/2/25: 32: 0/0/0
lpar25: 25: 560/U78A9.001.0123456-P1-C4/0x21010230/2/25: 32: 0/0/0
lpar29: 29: 568/U78A9.001.0123456-P1-C2/0x21010238/2/29: 32: 0/0/0
lpar5: 5: 521/U78A9.001.0123456-P1-C13/0x21010209/2/5: 32: 0/3/3
lpar5: 5: 520/U78A9.001.0123456-P1-C14/0x21010208/2/5: 32: 0/3/3
lpar9: 9: 529/U78A9.001.0123456-P1-C11/0x21010211/2/9: 32: 0/3/3
lpar9: 9: 528/U78A9.001.0123456-P1-C12/0x21010210/2/9: 32: 0/3/3
cec1: PendingPumpMode=1,CurrentPumpMode=1,OctantCount=8:
OctantID=0,PendingOctCfg=5,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=1,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=2,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=3,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=4,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=5,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=6,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
OctantID=7,PendingOctCfg=1,CurrentOctCfg=1,PendingMemoryInterleaveMode=2,
→CurrentMemoryInterleaveMode=2;
Number of BSR arrays: 256,Bytes per BSR array: 4096,Available BSR array: 0;
Available huge page memory(in pages):     0
Configurable huge page memory(in pages):  12
Page Size(in GB):                         16
Maximum huge page memory(in pages):       24
Requested huge page memory(in pages):     15
Number of BSR arrays: 256,Bytes per BSR array: 4096,Available BSR array: 0;
Available huge page memory(in pages):     0
Configurable huge page memory(in pages):  12
Page Size(in GB):                         16
Maximum huge page memory(in pages):       24
Requested huge page memory(in pages):     15
```

5. To list the virtual machine's directory entry:

```
lsvm gpok3
```

Output is similar to:

```
gpok3: USER LNX3 PWD 512M 1G G
gpok3: INCLUDE LNXDFLT
gpok3: COMMAND SET VSWITCH VSW2 GRANT LNX3
```

6. For DFM-managed normal power machine, list out the detailed resource information:

```
lsvm cec
```

Output is similar to:

```
cec: HYP Configurable Processors: 16, Avail Processors: 16.
HYP Configurable Memory:32.00 GB(128 regions).
HYP Available Memory:   31.25 GB(125 regions).
HYP Memory Region Size: 0.25 GB(256 MB).
cec: All Physical I/O info:
65535,519,U78AA.001.WZSGVU7-P1-C7,0x21010207,0xffff(Empty Slot)
65535,518,U78AA.001.WZSGVU7-P1-C6,0x21010206,0xffff(Empty Slot)
65535,517,U78AA.001.WZSGVU7-P1-C5,0x21010205,0xffff(Empty Slot)
65535,516,U78AA.001.WZSGVU7-P1-C4,0x21010204,0xffff(Empty Slot)
65535,514,U78AA.001.WZSGVU7-P1-C19,0x21010202,0xffff(Empty Slot)
65535,513,U78AA.001.WZSGVU7-P1-T7,0x21010201,0xc03(USB Controller)
65535,512,U78AA.001.WZSGVU7-P1-T9,0x21010200,0x104(RAID Controller)
cec: Huge Page Memory
Available huge page memory(in pages):    2
Configurable huge page memory(in pages): 2
Page Size(in GB):                        16
Maximum huge page memory(in pages):      4
Requested huge page memory(in pages):    2
cec: Barrier Synchronization Register(BSR)
Number of BSR arrays: 256
Bytes per BSR array:  4096
Available BSR array:  256
```

Note: The lines list in "All Physical I/O info" section represent all the physical I/O resource information. The format is like "owner_lparid,slot_id,physical resource name,drc_index,slot_class_code(class discription)". The 'drc index' is short for Dynamic Resource Configuration Index, it uniquely indicate a physical I/O resource in normal power machine.

For DFM-managed partition on normal power machine, list out the detailed information:

```
lsvm lpar1
```

Output is similar to:

```
lpar1: Lpar Processor Info:
Curr Processor Min: 1.
Curr Processor Req: 16.
Curr Processor Max: 16.
lpar1: Lpar Memory Info:
Curr Memory Min: 0.25 GB(1 regions).
Curr Memory Req: 30.75 GB(123 regions).
Curr Memory Max: 32.00 GB(128 regions).
lpar1: 1,519,U78AA.001.WZSGVU7-P1-C7,0x21010207,0xffff(Empty Slot)
lpar1: 1,518,U78AA.001.WZSGVU7-P1-C6,0x21010206,0xffff(Empty Slot)
lpar1: 1,517,U78AA.001.WZSGVU7-P1-C5,0x21010205,0xffff(Empty Slot)
lpar1: 1,516,U78AA.001.WZSGVU7-P1-C4,0x21010204,0xffff(Empty Slot)
lpar1: 1,514,U78AA.001.WZSGVU7-P1-C19,0x21010202,0xffff(Empty Slot)
lpar1: 1,513,U78AA.001.WZSGVU7-P1-T7,0x21010201,0xc03(USB Controller)
lpar1: 1,512,U78AA.001.WZSGVU7-P1-T9,0x21010200,0x104(RAID Controller)
lpar1: 1/2/2
lpar1: 256.
```

FILES

/opt/xcat/bin/lsvm

SEE ALSO

mkvm(1)|mkvm.1, chvm(1)|chvm.1, rmvm(1)|rmvm.1

lsxcatd.1

NAME

lsxcatd - lists xCAT daemon information.

SYNOPSIS

lsxcatd [**-h** | **--help** | **-v** | **--version** | **-d** | **--database** |**-t** | **--nodetype** | **-a** | **--all**]

DESCRIPTION

The **lsxcat** command lists important xCAT daemon (xcatd) information.

OPTIONS

-v|--version

> Command Version.

-h|--help

> Display usage message.

-d|--database

> Displays information about the current database being used by xCAT.

-t|--nodetype

> Displays whether the node is a Management Node or a Service Node.

-a|--all

> Displays all information about the daemon supported by the command.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To display information about the current database:

```
lsxcatd -d
```

Output is similar to:

```
cfgloc=Pg:dbname=xcatdb;host=7.777.47.250|xcatadm
dbengine=Pg
dbname=xcatdb
dbhost=7.777.47.250
dbadmin=xcatadm
```

2. To display all information:

```
lsxcatd -a
```

Output is similar to:

Version 2.8.5 (git commit 0d4888af5a7a96ed521cb0e32e2c918a9d13d7cc, built Tue Jul 29 02:22:47 EDT 2014) This is a Management Node cfgloc=mysql:dbname=xcatdb;host=9.114.34.44|xcatadmin dbengine=mysql dbname=xcatdb dbhost=9.114.34.44 dbadmin=xcatadmin

FILES

/opt/xcat/bin/lsxcatd

SEE ALSO

mkdef.1

NAME

mkdef - Use this command to create xCAT data object definitions.

SYNOPSIS

mkdef [**-h** | **--help**] [**-t** *object-types*]

mkdef [**-V** | **--verbose**] [**-t** *object-types*] [**-o** *object-names*] [**-z** | **--stanza**] [**-d** | **--dynamic**] [**-f** | **--force**] [[**-w** *attr==val*] [**-w** *attr=~val*] ...] [*noderange*] [*attr=val* [*attr=val...*]]

[**-u** **provmethod**=<*install*|*netboot*|*statelite*> **profile**=<xxx> [*osvers=value*] [*osarch=value*]]

DESCRIPTION

This command is used to create xCAT object definitions which are stored in the xCAT database. If the definition already exists it will return an error message. The force option may be used to re-create a definition. In this case the old definition will be remove and the new definition will be created.

OPTIONS

attr=val [attr=val ...]

> Specifies one or more "attribute equals value" pairs, separated by spaces. Attr=val pairs must be specified last on the command line. Use the help option to get a list of valid attributes for each object type.

> Note: when creating node object definitions, the 'groups' attribute is required.

-d|--dynamic

> Use the dynamic option to create dynamic node groups. This option must be used with -w option.

-f|--force

> Use the force option to re-create object definitions. This option removes the old definition before creating the new one.

-h|--help

> Display usage message.

noderange

> A set of comma delimited node names and/or group names. (must be the first parameter) See the "noderange" man page for details on supported formats.

-o *object-names*

> A set of comma delimited object names.

-t *object-types*

> A set of comma delimited object types. Use the help option to get a list of valid object types.

-V|--verbose

> Verbose mode.

-w *attr==val* **-w** *attr=~val ...*

> Use one or multiple -w flags to specify the selection string that can be used to select objects. The operators ==, !=, =~ and !~ are available. For mkdef commmand, the -w flag only makes sense for creating dynamic node group. Use the help option to get a list of valid attributes for each object type.

> **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.

> Note: if the "val" fields includes spaces or any other characters that will be parsed by shell, the "attr<operator>val" needs to be quoted. If the operator is "!~", the "attr<operator>val" needs to be quoted using single quote.

-z|--stanza

> Indicates that the file being piped to the command is in stanza format. See the xcatstanzafile man page for details on using xCAT stanza files.

-u

> Fill in the attributes such as template file, pkglist file and otherpkglist file of osimage object based on the specified parameters. It will search "/install/custom/" directory first, and then "/opt/xcat/share/". The *provmethod* and *profile* must be specified. If *osvers* or *osarch* is not specified, the corresponding value of the management node will be used.

Note: this option only works for objtype **osimage**.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To create a site definition.

```
mkdef -t site -o clustersite  installdir=/xcatinstall
```

2. To create a basic node definition.

```
mkdef -t node -o node01 groups="all,aix"
```

3. To re-create the current definition of "node01".

```
mkdef -f -t node -o node01  nodetype=osi groups="linux"
```

(The group definitions are also created if they don't already exist.)

4. To create a set of different types of definitions based on information contained in a stanza file.

```
cat defstanzafile | mkdef -z
```

5. To create a group definition called LinuxNodes containing the nodes clstrn01 and clstrn02.

```
mkdef -t group -o LinuxNodes members="clstrn01,clstrn02"
```

6. To create a node definition for an FSP node using the attributes provided by the group fspnodes.

```
mkdef -t node fspn1 groups=fspnodes nodetype=fsp
```

7. To create node definitions for a set of node host names contained in the node range "node1,node2,node3"

```
mkdef -t node node1,node2,node3 power=hmc groups="all,aix"
```

8. To create a dynamic node group definition called HMCMgtNodes containing all the HMC managed nodes"

```
mkdef -t group -o HMCMgtNodes -d -w mgt==hmc -w cons==hmc
```

9. To create a dynamic node group definition called SLESNodes containing all the SLES nodes

```
mkdef -t group -o SLESNodes -d -w "os=~^sles[0-9]+$"
```

10. To create a entry (7.0) in the policy table for user admin1

```
mkdef -t policy -o 7.0 name=admin1 rule=allow
```

11. To create a node definition with nic attributes

```
mkdef -t node cn1 groups=all nicips.eth0="1.1.1.1|1.2.1.1" nicnetworks.eth0=
→"net1|net2" nictypes.eth0="Ethernet"
```

12. To create an osimage definition and fill in attributes automatically.

```
mkdef redhat6img -u profile=compute provmethod=statelite
```

FILES

$XCATROOT/bin/mkdef

(The XCATROOT environment variable is set when xCAT is installed. The default value is "/opt/xcat".)

NOTES

This command is part of the xCAT software product.

SEE ALSO

chdef(1)|chdef.1, lsdef(1)|lsdef.1, rmdef(1)|rmdef.1, xcatstanzafile(5)|xcatstanzafile.5

mkdsklsnode.1

NAME

mkdsklsnode - Use this xCAT command to define and initialize AIX/NIM diskless machines.

SYNOPSIS

mkdsklsnode [-h|--help]

mkdsklsnode [-V|--verbose] [-f|--force] [-n|--newname] [-i osimage_name] [-l location] [-u|--updateSN] [-k|--skipsync] [-p|--primarySN] [-b|--backupSN] [-S|--setuphanfs] noderange [attr=val [attr=val ...]]

DESCRIPTION

This xCAT command can be used to define and/or initialize AIX/NIM diskless machines. Once this step is completed you can use either the xCAT **rnetboot** command or the **rbootseq/rpower** commands to initiate a network boot of the nodes.

The command can be used to define and initialize a new NIM machine object or it can be used to reinitialize an existing machine to use a different operating system image.

This command will also create a NIM resolv_conf resource to be used when installing the node. If a resolv_conf resource is not already included in the xCAT osimage definition and if the "domain" and "nameservers" values are set then a new NIM resolv_conf resource will be created and allocated to the nodes.

The "domain" and "nameservers" attributes can be set in either the xCAT "network" definition used by the nodes or in the xCAT cluster "site" definition. The setting in the "network" definition will take priority.

The "search" field of the resolv.conf file will contain a list all the domains listed in the xCAT network definitions and the xCAT site definiton.

The "nameservers" value can either be set to a specific IP address or the "<xcatmaster>" key word. The "<xcatmaster>" key word means that the value of the "xcatmaster" attribute of the node definition will be used in the /etc/resolv.conf file. (I.e. The name of the install server as known by the node.)

You can set the "domain" and "nameservers" attributes by using the **chdef** command. For example:

chdef -t network -o clstr_net domain=cluster.com nameservers=<xcatmaster>

If the "domain" and "nameservers" attributes are not set in either the nodes "network" definition or the "site" definition then no new NIM resolv_conf resource will be created.

If you are using xCAT service nodes the **mkdsklsnode** command will automatically determine the correct server(s) for the node and create the NIM definitions on that server(s).

When creating a new NIM machine definition the default is to use the same name as the xCAT node name that is provided.

You can use the "-n" option of the mkdsklsnode command to create and initialize an alternate NIM machine definition for the same physical nodes. This option allows you to set up a new image to use when a node is next rebooted while the node is currently running. This is possible because the NIM name for a machine definition does not have to be the hostname of the node. This allows you to have multiple NIM machine definitions for the same physical node. The naming convention for the new NIM machine name is "<xcat_node_name>_<image_name>", (Ex. "node01_61spot"). Since all the NIM initialization can be done while the node is running the downtime for for the node is reduced to the time it takes to reboot.

Note: When using the "-n" option make sure that the new osimage you specify and all the NIM resources that are used are different than what are currently being used on the nodes. The NIM resources should not be shared between the old osimage and the new osimage.

You can use the force option to reinitialize a node if it already has resources allocated or it is in the wrong NIM state. This option will reset the NIM node and deallocate resources before reinititializing. Use this option with caution since reinitializing a node will stop the node if it is currently running.

After the mkdsklsnode command completes you can use the **lsnim** command to check the NIM node definition to see if it is ready for booting the node. ("lsnim -l <nim_node_name>").

You can supply your own scripts to be run on the management node or on the service node (if their is hierarchy) for a node during the **mkdsklsnode** command. Such scripts are called **prescripts**. They should be copied to /install/prescripts dirctory. A table called *prescripts* is used to specify the scripts and their associated actions. The scripts to be run at the beginning of the **mkdsklsnode** command are stored in the 'begin' column of *prescripts* table. The scripts to be run at the end of the **mkdsklsnode** command are stored in the 'end' column of *prescripts* table. Please run 'tabdump prescripts -d' command for details. An example for the 'begin' or the 'end' column is: *diskless:myscript1,myscript2*. The following two environment variables will be passed to each script: NODES contains all the names of the nodes that need to run the script for and ACTION contains the current current nodeset action, in this case "diskless". 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 will be invoked at at a time. If it is not specified, the script will be invoked once for all the nodes.

OPTIONS

attr=val [attr=val ...]

> Specifies one or more "attribute equals value" pairs, separated by spaces. Attr= val pairs must be specified last on the command line. These are used to specify additional values that can be passed to the underlying NIM commands.

> Valid values:

> **duplex**

Specifies the duplex setting (optional). Used when defining the NIM machine. Use this setting to configure the client's network interface. This value can be full or half. The default is full. (ex. "duplex=full")

speed

Specifies the speed setting (optional). Used when defining the NIM machine. This is the communication speed to use when configuring the client's network interface. This value can be 10, 100, or 1000. The default is 100. (ex. "speed=100")

psize

Specifies the size in Megabytes of the paging space for the diskless node.(optional) Used when initializing the NIM machine. The minimum and default size is 64 MB of paging space. (ex. "psize=256")

sparse_paging

Specifies that the paging file should be created as an AIX sparse file, (ex. "sparse_paging=yes"). The default is "no".

dump_iscsi_port

The tcpip port number to use to communicate dump images from the client to the dump resource server. Normally set by default. This port number is used by a dump resource server.

configdump

Specifies the type dump to be collected from the client. The values are "selective", "full", and "none". If the configdump attribute is set to "full" or "selective" the client will automatically be configured to dump to an iSCSI target device. The "selective" memory dump will avoid dumping user data. The "full" memory dump will dump all the memory of the client partition. Selective and full memory dumps will be stored in subdirectory of the dump resource allocated to the client. This attribute is saved in the xCAT osimage definition.

-b |--backupSN

When using backup service nodes only update the backup. The default is to update both the primary and backup service nodes.

-f |--force

Use the force option to reinitialize the NIM machines.

-h |--help

Display usage message.

-i image_name

The name of an existing xCAT osimage definition. If this information is not provided on the command line the code checks the node definition for the value of the "provmethod" attribute. If the "-i" value is provided on the command line then that value will be used to set the "provmethod" attribute of the node definitions.

-k|--skipsync

Use this option to have the mkdsklsnode command skip the NIM sync_roots operation. This option should only be used if you are certain that the shared_root resource does not have to be updated from the SPOT. Normally, when the SPOT is updated, you should do a sync_roots on the shared_root resource.

-l|--location

The directory location to use when creating new NIM resolv_conf resources. The default location is /install/nim.

-n|--newname

> Create a new NIM machine object name for the xCAT node. Use the naming convention "<xcat_node_name>_<image_name>" for the new NIM machine definition.

-p|--primarySN

> When using backup service nodes only update the primary. The default is to update both the primary and backup service nodes.

-S|--setuphanfs

> Setup NFSv4 replication between the primary service nodes and backup service nodes to provide high availability NFS for the compute nodes. This option only exports the /install directory with NFSv4 replication settings, the data synchronization between the primary service nodes and backup service nodes needs to be taken care of through some mechanism.

-u|--updateSN

> Use this option if you wish to update the osimages but do not want to define or initialize the NIM client definitions. This option is only valid when the xCAT "site" definition attribute "sharedinstall" is set to either "sns" or "all".

noderange

> A set of comma delimited node names and/or group names. See the "noderange" man page for details on additional supported formats.

-V |--verbose

> Verbose mode.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

1

> Initialize an xCAT node named "node01" as an AIX diskless machine. The xCAT osimage named "61spot" should be used to boot the node.

> **mkdsklsnode -i 61spot node01**

2

> Initialize all AIX diskless nodes contained in the xCAT node group called "aixnodes" using the image definitions pointed to by the "provmethod" attribute of the xCAT node definitions.

> **mkdsklsnode aixnodes**

3

Initialize diskless node "clstrn29" using the xCAT osimage called "61dskls". Also set the paging size to be 128M and specify the paging file be an AIX sparse file.

mkdsklsnode -i 61dskls clstrn29 psize=128 sparse_paging=yes

4

Initialize an xCAT node called "node02" as an AIX diskless node. Create a new NIM machine definition name with the osimage as an extension to the xCAT node name.

mkdsklsnode -n -i 61spot node02

FILES

/opt/xcat/bin/mkdsklsnode

NOTES

This command is part of the xCAT software product.

SEE ALSO

rmdsklsnode(1)|rmdsklsnode.1

mkflexnode.1

NAME

mkflexnode - Create a flexible node.

SYNOPSIS

mkflexnode [-h | -****-help]

mkflexnode [-v | -****-version]

mkflexnode *noderange*

DESCRIPTION

A flexible node is a **Partition** in a complex. Creating a flexible node is to create a partition which including all the slots defined in the xCAT blade node.

Before creating a flexible node, a general xCAT blade node should be defined. The *id* attribute of this node should be a node range like 'a-b', it means the blades installed in slots 'a-b' need to be assigned to the partition. 'a' is the start slot, 'b' is the end slot. If this partition only have one slot, the slot range can be 'a'.

The action of creating flexible node will impact the hardware status. Before creating it, the blades in the slot range should be in **power off** state.

After the creating, use the **lsflexnode** to check the status of the node.

The *noderange* only can be a blade node.

OPTIONS

-h | --help

Display the usage message.

-v | --version

Display the version information.

EXAMPLES

1

Create a flexible node base on the xCAT node blade1.

The blade1 should belong to a complex, the *id* attribute should be set correctly and all the slots should be in **power off** state.

```
mkflexnode blade1
```

FILES

/opt/xcat/bin/mkflexnode

SEE ALSO

lsflexnode(1)|lsflexnode.1, rmflexnode(1)|rmflexnode.1

mkhwconn.1

NAME

mkhwconn - Sets up connections for CEC and Frame nodes to HMC nodes or hardware server.

SYNOPSIS

mkhwconn [**-h**| **--help**]

mkhwconn [**-v**| **--version**]

PPC (with HMC) specific:

mkhwconn [**-V**| **--verbose**] *noderange* **-t** [**--port** *port_value*]

mkhwconn [**-V**| **--verbose**] *noderange* **-s** [*hmcnode* **--port** *port_value*]

mkhwconn [**-V**| **--verbose**] *noderange* **-p** *hmc* [**-P** *passwd*] [**--port** *port_value*]

PPC (using Direct FSP Management) specific:

mkhwconn *noderange* **-t** [**-T tooltype**] [**--port** *port_value*]

DESCRIPTION

For PPC (with HMC) specific:

This command is used to set up connections for CEC and Frame nodes to HMC nodes. (If the connection already exists, it will not break it.) This command is useful when you have multiple HMCs, each of which will manage a subset of the CECs/Frames. Use **mkhwconn** to tell each HMC which CECs/Frames it should manage. When using this, you should turn off the self-discovery on each HMC. You also need to put all the HMCs and all the Frames on a single flat service network.

When **-t** is specified, this command reads the connection information from the xCAT ppc table (e.g. the parent attribute), and read the user/password from the ppcdirect table. Then this command will assign CEC nodes and Frame nodes to HMC nodes.

When **-p** is specified, this command gets the connection information from command line arguments. If **-P** is not specified, the default password for CEC and Frame nodes is used.

The flag **-s** is used to make the connection between the frame and its Service focal point(HMC). Makehwconn will also set the connections between the CECs within this Frame and the HMC. The sfp of the frame/CEC can either be defined in ppc table beforehand or specified in command line after the flag -s. If the user use mkhwconn noderange -s HMC_name, it will not only make the connections but also set the sfp attributes for these nodes in PPC table.

In any case, before running this command, the CEC and Frame nodes need be defined with correct nodetype.nodetype value (cec or frame) and nodehm.mgt value (hmc).

Note: If a CEC belongs to a frame, which has a BPA installed, this CEC should not be assigned to an HMC individually. Instead, the whole frame should be assigned to the HMC.

For PPC (using Direct FSP Management) specific:

It is used to set up connections for CEC and Frame node to Hardware Server on management node (or service node). It only could be done according to the node definition in xCAT DB. And this command will try to read the user/password from the ppcdirect table first. If fails, then read them from passwd table. Commonly , the username is **HMC**. If using the **ppcdirect** table, each CEC/Frame and user/password should be stored in **ppcdirect** table. If using the **passwd** table, the key should be "**cec**" or "**frame**", and the related user/password are stored in **passwd** table.

When **--port** is specified, this command will create the connections for CECs/Frames whose side in **vpd** table is equal to port value.

OPTIONS

-h|--help

> Display usage message.

-t

> Read connection information from xCAT DB (ppc and ppcdirect tables). Use this option if you need to connect multiple CECs/Frames to multiple HMCs in a single command.

-p

> The HMC node name. Only one HMC nodes can be specified by this flag. To setup connection for multiple HMC nodes, use flag **-t**.

-P

> The password of HMC based CEC/Frame login user(Default user name is 'HMC'). This flag is optional.

-T

> The tooltype is used to communicate to the CEC/Frame. The value could be **lpar** or **fnm**. The tooltype value **lpar** is for xCAT and **fnm** is for CNM. The default value is "**lpar**".

--port

> The port value specifies which special side will be used to create the connection to the CEC/Frame. The value could only be specified as "**0**" or "**1**" and the default value is "**0,1**". If the user wants to use all ports to create the connection, he should not specify this value. If the port value is specified as "**0**", in the vpd table, the side column should be **A-0** and **B-0**; If the port value is specified as "**1**", the side column should be **A-1** and **B-1**. When making hardware connection between CEC/Frame and HMC, the value is used to specify the fsp/bpa port of the cec/frame and will be organized in order of "**A-0,A-1,B-0,B-1**". If any side does not exist, the side would simply be ignored. Generally, only one port of a fsp/bap can be connected while another port be used as backup.

-s

> The flag -s is used to make the connection between the frame and its Service Focal Point(HMC). -s flag is not supposed to work with other functional flags.

-V|--verbose

> Verbose output.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To setup the connection for all CEC nodes in node group cec to HMC node, according to the definition in xCAT DB:

```
mkhwconn cec -t
```

2. To setup the connection for Frame nodes in node group frame to HMC node hmc1, with password 'abc123':

```
mkhwconn frame -p hmc1 -P abc123
```

3. To setup the connections for all CEC nodes in node group cec to hardware server, and the tooltype value is lpar:

```
mkhwconn cec -t -T lpar
```

4. To setup the connections for all cecs nodes in node group cec to hardware server, and the tooltype value is lpar, and the port value is 1:

```
mkhwconn cec -t -T lpar --port 1
```

5. To setup the connection between the frame and it's SFP node. This command will also set the connections between the CECs within this frame and their SFP node. User need to define HMC_name in the database in

advance, but no need to set the sfp attribute for these node, xCAT will set the HMC_name as ppc.sfp for these nodes. The CECs within this frame should have the same sfp attribute as the frame.

```
mkhwconn cec -s HMC_name -P HMC_passwd
```

FILES

$XCATROOT/bin/mkhwconn

(The XCATROOT environment variable is set when xCAT is installed. The default value is "/opt/xcat".)

NOTES

This command is part of the xCAT software product.

SEE ALSO

lshwconn(1)|lshwconn.1, rmhwconn(1)|rmhwconn.1

mknimimage.1

NAME

mknimimage - Use this xCAT command to create xCAT osimage definitions and related AIX/NIM resources. The command can also be used to update an existing AIX diskless image(SPOT).

SYNOPSIS

mknimimage [-h | --help]

mknimimage [-V] -u osimage_name [attr=val [attr=val ...]]

mknimimage [-V] [-f|--force] [-r|--sharedroot] [-D|--mkdumpres] [-l location] [-c|--completeosimage] [-s image_source] [-i current_image] [-p|--cplpp] [-t nimtype] [-m nimmethod] [-n mksysbnode] [-b mksysbfile] osimage_name [attr=val [attr=val ...]]

DESCRIPTION

This command will create both an xCAT osimage definition and the corresponding NIM resource definitions. The command can also be used to update an existing AIX diskless image(SPOT).

The command will also install the NIM master software and configure NIM if needed.

The naming convention for the NIM SPOT resource definition is to use the same name as the xCAT osimage. The naming convention for any other NIM resources that are created is "<osimage_name>_<resource_type>". (ex. "61image_lpp_source")

When creating a mksysb image definition you must specify either the "-n" or the "-b" option. The "-n" option can be used to create a mksysb image from an existing NIM client machine. The "-b" option can be used to specify an existing mksysb backup file.

Adding software and configuration files to the osimage.

When creating a diskless osimage definition you also have the option of automatically updating the NIM SPOT resource. You can have additional software installed or you can have configuration files added or updated. To have software installed you must provide either the names of NIM installp_bundle resources or fileset names on the command line using the "attr=val" option. You may also supply the installp flags, RPM flags, emgr flags to use when installing the software.

To have configuration files updated you must provide the full path name of a "synclists" file which contains the the list of actual files to update. The xCAT osimage definition that is created will contain the installp_bundle, otherpkgs, and synclists files that are provided on the command line.

Updating an existing xCAT osimage

If you wish to update an existing diskless image after it has already been created you can use the "-u" (update) option. In this case the xCAT osimage definition will not be updated.

There are two ways to use the update feature.

You can update the osimage definition and run the **mknimimage** command with no "installp_bundle", "otherpkgs", or "synclists" command line values. The information for updating the SPOT will come from the osimage definition only. This has the advantage of keeping a record of any changes that were made to the SPOT.

Or, you could do a more ad hoc update by providing one or more of the "installp_bundle", "otherpkgs", or "synclists" values on the command line. If any of these values are provided the **mknimimage** command will use those values only. The osimage definition will not be used or updated.

WARNING: Installing random RPM packages in a SPOT may have unpredictable consequences. The SPOT is a very restricted environment and some RPM packages may corrupt the SPOT or even hang your management system. Try to be very careful about the packages you install. When installing RPMs, if the mknimimage command hangs or if there are file systems left mounted after the command completes you may need to reboot your management node to recover. This is a limitation of the current AIX support for diskless systems

Copying an xCAT osimage.

You can use the "-i" and "-p" options to copy an existing diskless osimage. To do this you must supply the name of an existing xCAT osimage definition and the name of the new osimage you wish to create. The **mknimimage** command will do the following:

- create a new xCAT osimage definition using the new name that was specified.

- copy the NIM SPOT resource to a new location and define it to NIM using a new name.

- if the original osimage included a NIM "shared_root" resource then a new shared_root resource will be created for the new SPOT.

- any other resources (or attributes) included in the original osimage will be included in the new osimage definition.

- if the "-p" option is specified then the original NIM lpp_source resource will be copied to a new location and redefined to NIM. (The default would be to use the original lpp_source - to save file system space.)

Additional information

IMPORTANT: The NIM lpp_source and SPOT resources can get quite large. Always make sure that you have sufficient file system space available before running the **mknimimage** command.

To list the contents of the xCAT osimage definition use the xCAT **lsdef** command ("lsdef -t osimage -l -o <osimage_name>").

To check the validity of a SPOT or lpp_source resource

To remove an xCAT osimage definition along with the associated NIM resource definitions use the **rmnimimage** command. Be careful not to accidently remove NIM resources if they are still needed.

To list a NIM resource definition use the AIX **lsnim** command ("lsnim -l <resource_name>").

To check the validity of a SPOT or lpp_source resource use the AIX **nim** command ("nim -o check <resourec-name>").

To remove specific NIM resource definitons use the AIX **nim** command. ("nim -o remove <resource-name>").

OPTIONS

attr=val [attr=val ...]

> Specifies one or more "attribute equals value" pairs, separated by spaces. Attr=val pairs must be specified last on the command line.

> Currently supported attributes:

> **bosinst_data**

>> The name of a NIM bosinst_data resource.

> **dump**

>> The name of the NIM dump resource.

> **fb_script**

>> The name of a NIM fb_script resource.

> **home**

>> The name of the NIM home resource.

> **installp_bundle**

>> One or more comma separated NIM installp_bundle resources.

> **lpp_source**

>> The name of the NIM lpp_source resource.

> **mksysb**

>> The name of a NIM mksysb resource.

> **otherpkgs**

>> One or more comma separated installp, emgr, or rpm packages. The packages must have prefixes of 'I:', 'E:', or 'R:', respectively. (ex. R:foo.rpm)

> **paging**

>> The name of the NIM paging resource.

> **resolv_conf**

>> The name of the NIM resolv_conf resource.

> **root**

>> The name of the NIM root resource.

> **script**

>> The name of a NIM script resource.

> **shared_home**

>> The name of the NIM shared_home resource.

shared_root

A shared_root resource represents a directory that can be used as a / (root) directory by one or more diskless clients.

spot

The name of the NIM SPOT resource.

synclists

The fully qualified name of a file containing a list of files to synchronize on the nodes.

tmp

The name of the NIM tmp resource.

installp_flags

The alternate flags to be passed along to the AIX installp command. (The default for installp_flags is "-abgQXY".)

rpm_flags

The alternate flags to be passed along to the AIX rpm command. (The default for rpm_flags is "-Uvh ".) The mknimimage command will check each rpm to see if it is installed. It will not be reinstalled unless you specify the appropriate rpm option, such as '–replacepkgs'.

emgr_flags

The alternate flags to be passed along to the AIX emgr command. (There is no default flags for the emgr command.)

dumpsize

The maximum size for a single dump image the dump resource will accept. Space is not allocated until a client starts to dump. The default size is 50GB. The dump resource should be large enough to hold the expected AIX dump and snap data.

max_dumps

The maximum number of archived dumps for an individual client. The default is one.

snapcollect

Indicates that after a dump is collected then snap data should be collected. The snap data will be collected in the clients dump resource directory. Values are "yes" or "no". The default is "no".

nfs_vers

Value Specifies the NFS protocol version required for NFS access.

nfs_sec

Value Specifies the security method required for NFS access.

Note that you may specify multiple "script", "otherpkgs", and "installp_bundle" resources by using a comma seperated list. (ex. "script=ascript,bscript"). RPM names may be included in the "otherpkgs" list by using a "R:" prefix(ex. "R:whatever.rpm"). epkg (AIX interim fix package) file names may be included in the "otherpkgs" using the 'E:' prefix. (ex. "otherpkgs=E:IZ38930TL0.120304.epkg.Z").

-b mksysbfile

Used to specify the path name of a mksysb file to use when defining a NIM mksysb resource.

-c|--completeosimage

Complete the creation of the osimage definition passed in on the command line. This option will use any additonal values passed in on the command line and/or it will attempt to create required resources in order to complete the definition of the xCAT osimage. For example, if the osimage definition is missing a spot or shared_root resource the command will create those resources and add them to the osimage definition.

-f|--force

Use the force option to re-create xCAT osimage definition. This option removes the old definition before creating the new one. It does not remove any of the NIM resource definitions named in the osimage definition. Use the **rmnimimage** command to remove the NIM resources associated with an xCAT osimage definition.

-h |--help

Display usage message.

osimage_name

The name of the xCAT osimage definition. This will be used as the name of the xCAT osimage definition as well as the name of the NIM SPOT resource.

-D|--mkdumpres

Create a diskless dump resource.

-i current_image

The name of an existing xCAT osimage that should be copied to make a new xCAT osimage definition. Only valid when defining a "diskless" or "dataless" type image.

-l location

The directory location to use when creating new NIM resources. The default location is /install/nim.

-m nimmethod

Used to specify the NIM installation method to use. The possible values are "rte" and "mksysb". The default is "rte".

-n mksysbnode

The xCAT node to use to create a mksysb image. The node must be a defined as a NIM client machine.

-p|--cplpp

Use this option when copying existing diskless osimages to indicate that you also wish to have the lpp_resource copied. This option is only valid when using the "-i" option.

-r|--sharedroot

Use this option to specify that a NIM "shared_root" resource be created for the AIX diskless nodes. The default is to create a NIM "root" resource. This feature is only available when using AIX version 6.1.4 or beyond. See the AIX/NIM documentation for a description of the "root" and "shared_root" resources.

-s image_source

The source of software to use when creating the new NIM lpp_source resource. This could be a source directory or a previously defined NIM lpp_source resource name.

-t nimtype

Used to specify the NIM machine type. The possible values are "standalone", "diskless" or "dataless". The default is "standalone".

-u

Used to update an AIX/NIM SPOT resource with additional software and configuration files. This option is only valid for xCAT diskless osimage objects. The SPOT resource associated with the xCAT osimage definition will be updated. This option can also be used to update the nfs_vers attribute from NFSv3 to NFSv4 for the NIM resources associated with diskful or diskless image.

-V |--verbose

Verbose mode.

RETURN VALUE

0

The command completed successfully.

1

An error has occurred.

EXAMPLES

1. Create an osimage definition and the basic NIM resources needed to do a NIM "standalone" "rte" installation of node "node01". Assume the software contained on the AIX product media has been copied to the /AIX/instimages directory.

mknimimage -s /AIX/instimages 61image

2. Create an osimage definition that includes some additional NIM resources.

mknimimage -s /AIX/instimages 61image installp_bundle=mybndlres,addswbnd

This command will create lpp_source, spot, and bosinst_data resources using the source specified by the "-s" option. The installp_bundle information will also be included in the osimage definition. The mybndlres and addswbnd resources must be created before using this osimage definition to install a node.

3. Create an osimage definition that includes a mksysb image and related resources.

mknimimage -m mksysb -n node27 newsysb spot=myspot bosinst_data=mybdata

This command will use node27 to create a mksysb backup image and use that to define a NIM mksysb resource. The osimage definition will contain the name of the mksysb resource as well as the spot and bosinst_data resource.

4. Create an osimage definition using a mksysb image provided on the command line.

mknimimage -m mksysb -b /tmp/backups/mysysbimage newsysb spot=myspot bosinst_data=mybdata

This command defines a NIM mksysb resource using mysysbimage.

5. Create an osimage definition and create the required spot definition using the mksysb backup file provided on the command line.

mknimimage -m mksysb -b /tmp/backups/mysysbimage newsysb bosinst_data=mybdata

This command defines a NIM mksysb resource and a spot definition using mysysbimage.

6. Create a diskless image called 61dskls using the AIX source files provided in the /AIX/instimages directory.

mknimimage -t diskless -s /AIX/instimages 61dskls

7. Create a diskless image called "614dskls" that includes a NIM "shared_root" and a "dump" resource. Use the existing NIM lpp_resource called "614_lpp_source". Also specify verbose output.

---

mknimimage -V -r -D -t diskless -s 614_lpp_source 614dskls snapcollect=yes

The "snapcollect" attribute specifies that AIX "snap" data should be include when a system dump is initiated.

8. Create a new diskless image by copying an existing image.

mknimimage -t diskless -i 61cosi 61cosi_updt1

Note: If you also wish to have the original lpp_source copied and defined use the -p option.

mknimimage -t diskless -i 61cosi -p 61cosi_updt1

9. Create a diskless image using an existing lpp_source resource named "61cosi_lpp_source" and include NIM tmp and home resources. This assumes that the "mytmp" and "myhome" NIM resources have already been created by using NIM commands.

mknimimage -t diskless -s 61cosi_lpp_source 611cosi tmp=mytmp home=myhome

10. Create a diskless image and update it with additional software using rpm flags and configuration files.

mknimimage -t diskless -s 61cosi_lpp_source 61dskls otherpkgs=I:fset1,R:foo.rpm,E:IZ38930TL0.120304.epkg.Z synclists=/install/mysyncfile rpm_flags="-i --nodeps"

The xCAT osimage definition created by this command will include the "otherpkgs" and "synclists" values. The NIM SPOT resource associated with this osimage will be updated with the additional software using rpm flags "-i –nodeps" and configuration files.

11. Update an existing diskless image (AIX/NIM SPOT) using the information saved in the xCAT "61dskls" osimage definition. Also specify verbose messages.

mknimimage -V -u 61dskls

12. Update an existing diskless image called "61dskls". Install the additional software specified in the NIM "bndres1" and "bndres2" installp_bundle resources using the installp flags "-agcQX". (The NIM "bndres1" and "bndres2" definitions must be created before using them in this command.)

mknimimage -u 61dskls installp_bundle=bndres1,bndres2 installp_flags="-agcQX"

Note that when "installp_bundle", "otherpkgs", or "synclists" values are specified with the "-u" option then the xCAT osimage definiton is not used or updated.

13. Update an existing image to support NFSv4. Also specify verbose messages.

mknimimage -V -u 61dskls nfs_vers=4

FILES

/opt/xcat/bin/mknimimage

NOTES

This command is part of the xCAT software product.

SEE ALSO

rmnimimage(1)|rmnimimage.1

---

mkvlan.1

NAME

mkvlan - It takes a list of nodes and create a private tagged vlan for them.

SYNOPSIS

mkvlan [*vlanid*] **-n | --nodes** *noderange* [**-t | --net** *subnet*] [**-m | --mask** *netmask*] [**-p | --prefix** *hostname_prefix*] [**-i | --interface** *nic*]

mkvlan [**-h | --help**]

mkvlan [**-v | --version**]

DESCRIPTION

The **mkvlan** command takes a list of nodes and move them to a private vlan.

This command will configure the switch to create a new tagged vlan on the given nic. The primary nic will be used if the nic is not specified. The new vlan ID is given by the command. However, if it is omitted, xCAT will automatically generate the new vlan ID by querying all the switches involved and finding out the smallest common number that is not used by any existing vlans. The subnet and the netmask for the vlan will be derived from the value of "vlannets" and "vlanmasks" from the *site* table if -t and -m are not specified. The following are the default site table entires:

```
vlannets="|(\d+)|10.($1+0).0.0|";
vlanmask="255.255.0.0";
```

The vlan network will be entered in the *networks* table. The nodes will be added to the vlan using the vlan tagging technique. And the new IP addresses and new hostnames will be assigned to the nodes. The -p flag specifies the node hostname prefix for the nodes. If it is not specified, by default, the hostnames for the nodes are having the following format:

v<vlanid>nY where Y is the node number. For example, the hostname for node 5 on vlan 10 is v10n5.

The *switch.vlan* will be updated with the new vlan id for the node for standaline nodes. For KVM guests, the *vm.nics* identifies which vlan this node belongs to. For example: vl3 means this node is in vlan 3.

If there are more than one switches involved in the vlan, the ports that connect to the switches need to entered in *switches.linkports* with the following format:

```
<port numner>:switch,<port number>:switch....
```

For example:

```
"42:switch1,43:switch2"
```

This command will automatically configure the cross-over ports if the given nodes are on different switches.

For added security, the root guard and bpdu guard will be enabled for the ports in this vlan. However, the guards will not be disabled if the ports are removed from the vlan using chvlan or rmvlan commands. To disable them, you need to use the switch command line interface. Please refer to the switch command line interface manual to see how to disable the root guard and bpdu guard for a port.

Parameters

vlanid is a unique vlan number. If it is omitted, xCAT will automatically generate the new vlan ID by querying all the switches involved and finding out the smallest common number that is not used by any existing vlans. Use **lsvlan** to find out the existing vlan ids used by xCAT.

OPTIONS

-n|--nodes The nodes or groups to be included in the vlan. It can be stand alone nodes or KVM guests. It takes the noderange format. Please check the man page for noderange for details.

-t|--net The subnet for the vlan.

-m|--mask The netmask for the vlan

-p|--prefix The prefix the the new hostnames for the nodes in the vlan.

-i|--interface The interface name where the vlan will be tagged on. If omitted, the xCAT management network will be assumed. For FVM, this is the interface name on the host.

-h|--help Display usage message.

-v|--version The Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

To start, the xCAT switches and switches table needs to be filled with switch and port info for the nodes. For example, the swith table will look like this:

#node,switch,port,vlan,interface,comments,disable "node1","switch1","10",,,, "node1","switch2","1",,"eth1",, "node2","switch1","11",,"primary",, "node2","switch2","2",,"eth1",, "node3","switch1","12",,"primary:eth0",, "node3","switch2","3",,"eth1",,

Please note that the interface value for the management (primary) network can be empty, the word "primary" or "primary:ethx". For other networks, the interface attribute must be specified.

The following is an example of the switches table

#switch,snmpversion,username,password,privacy,auth,linkports,sshusername,sshpassword,switchtype,comments,disable "switch1","3","username","passw0rd",,"sha","48:switch2",,,,, "switch2","2",,,,,"43:switch1",,,,,

1. To make a private vlan for node1, node2 and node3

   ```
   mkvlan -n node1,node2,node3
   ```

   The vlan will be created on eth0 for the nodes.

2. To make a private vlan for node1, node2 and node3 on eth1,

   ```
   mkvlan -n node1,node2,node3 -i eth1
   ```

3. TO make a private vlan for node1, node2 with given subnet and netmask.

```
mkvlan -n node1,node2,node3 -t 10.3.2.0 -m 255.255.255.0
```

4. To make a private vlan for KVM guests node1 and node2

```
chtab key=usexhrm site.vlaue=1

mkdef node1 arch=x86_64 groups=kvm,all installnic=mac primarynic=mac mgt=kvm␣
↪netboot=pxe nfsserver=10.1.0.204 os=rhels6 profile=compute provmethod=install␣
↪serialport=0 serialspeed=115200 vmcpus=1 vmhost=x3650n01 vmmemory=512␣
↪vmnics=br0 vmstorage=nfs://10.1.0.203/vms

mkdef node2 arch=x86_64 groups=kvm,all installnic=mac primarynic=mac mgt=kvm␣
↪netboot=pxe nfsserver=10.1.0.204 os=rhels6 profile=compute provmethod=install␣
↪serialport=0 serialspeed=115200 vmcpus=1 vmhost=x3650n01 vmmemory=512␣
↪vmnics=br0 vmstorage=nfs://10.1.0.203/vms

mkvlan -n node1,node2

mkvm node1,node2 -s 20G

rpower node1,node2 on

rinstall node1,node2
```

FILES

/opt/xcat/bin/mkvlan

SEE ALSO

chvlan(1)|chvlan.1, rmvlan(1)|rmvlan.1, lsvlan(1)|lsvlan.1

mkvm.1

NAME

mkvm - Creates HMC-, DFM-, IVM-, and zVM-managed partitions or other virtual machines.

SYNOPSIS

Common:

mkvm [**-h**| **--help**]

mkvm [**-v**| **--version**]

For PPC (with HMC) specific:

mkvm [**-V**| **--verbose**] *noderange* **-i** *id* **-l** *singlenode*

mkvm [**-V**| **--verbose**] *noderange* **-c** *destcec* **-p** *profile*

mkvm [**-V**| **--verbose**] *noderange* **--full**

For PPC (using Direct FSP Management) specific:

mkvm *noderange* [**--full**]

mkvm *noderange* [**vmcpus=min/req/max**] [**vmmemory=min/req/max**] [**vmphyslots=drc_index1,drc_index2...**] [**vmothersetting=hugepage:N,bsr:N**] [**vmnics=vlan1[,vlan2..]**] [**vmstorage=<N|viosnode:slotid>**] [**--vios**]

For KVM:

mkvm *noderange* [**-m**|**--master** *mastername*] [**-s**|**--size** *disksize*] [**--mem** *memsize*] [**--cpus** *cpucount*] [**-f**|**--force**]

For Vmware:

mkvm *noderange* [**-s** | **--size** *disksize*] [**--mem** *memsize*] [**--cpus** *cpucount*]

For zVM:

mkvm *noderange* [*directory_entry_file_path*]

mkvm *noderange* [*source_virtual_machine*] [**pool=** *disk_pool*]

DESCRIPTION

For PPC (with HMC) specific:

The first form of mkvm command creates new partition(s) with the same profile/resources as the partition specified by *singlenode*. The -i and *noderange* specify the starting numeric partition number and the *noderange* for the newly created partitions, respectively. The LHEA port numbers and the HCA index numbers will be automatically increased if they are defined in the source partition.

The second form of this command duplicates all the partitions from the source specified by *profile* to the destination specified by *destcec*. The source and destination CECs can be managed by different HMCs.

Please make sure the nodes in the *noderange* is defined in the *nodelist* table and the *mgt* is set to 'hmc' in the *nodehm* table before running this command.

Please note that the mkvm command currently only supports creating standard LPARs, not virtual LPARs working with VIOS server.

For PPC (using Direct FSP Management) specific:

With option *full*, a partition using all the resources on a normal power machine will be created.

If no option is specified, a partition using the parameters specified with attributes such as 'vmcpus', 'vmmory', 'vm-physlots', 'vmothersetting', 'vmnics', 'vmstorage' will be created. Those attributes can either be specified with '*def' commands running before or be specified with this command.

For KVM and Vmware:

The mkvm command creates new virtual machine(s) with the *disksize* size of hard disk, *memsize* size of memory and *cpucount* number of cpu.

For KVM: If **-f | --force** is specified, the storage will be destroyed first if it existed.

For zVM:

The first form of mkvm creates a new virtual machine based on a directory entry.

The second form of this creates a new virtual machine with the same profile/resources as the specified node (cloning).

OPTIONS

-h|--help

> Display usage message.

-c

> The cec (fsp) name for the destination.

--cpus

> The cpu count which will be created for the kvm/vmware virtual machine.

--full

> Request to create a new full system partition for each CEC.

vmcpus=value vmmemory=value vmphyslots=value vmothersetting=value vmnics=value vmstorage=value [--vios]

> To specify the parameters which are used to create a partition. The *vmcpus*, *vmmemory* are necessay, and the value specified with this command have a more high priority. If the value of any of the three options is not specified, the corresponding value specified for the node object will be used. If any of the three attributes is neither specified with this command nor specified with the node object, error information will be returned. To reference to lsvm(1)|lsvm.1 for more information about 'drc_index' for *vmphyslots*.

> The option *vios* is used to specify the partition that will be created is a VIOS partition. If specified, the value for *vmstorage* shall be number which indicate the number of vSCSI server adapter will be created, and if no value specified for *vmphyslots*, all the physical slot of the power machine will be asigned to VIOS partition. If not specified, it shall be in form of *vios_name:server_slotid* to specify the vios and the virtual slot id of the vSCSI server adapter that will be connected from the Logical partition.

-f|--force

> If **-f|--force** is specified, the storage will be destroyed first if it existed.

-i

Starting numeric id of the newly created partitions.

-l

The partition name of the source.

--mem

The memory size which will be used for the new created kvm/vmware virtual machine. Unit is Megabyte.

-p

The file that contains the profiles for the source partitions.

-s|--size

The size of storage which will be created for the kvm/vmware virtual machine.

-v|--version

Command Version.

-V|--verbose

Verbose output.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To create a new HMC-managed partition lpar5 based on the profile/resources of lpar4, enter:

```
mkdef -t node -o lpar5 mgt=hmc groups=all
```

then:

```
mkvm lpar5 -i 5 -l lpar4
```

Output is similar to:

```
lpar5: Success
```

2. To create new HMC-managed partitions lpar5-lpar8 based on the profile/resources of lpar4, enter:

```
mkdef -t node -o lpar5-lpar8 mgt=hmc groups=all
```

then:

```
mkvm lpar5-lpar8 -i 5 -l lpar4
```

Output is similar to:

```
lpar5: Success
lpar6: Success
lpar7: Success
lpar8: Success
```

3. To duplicate all the HMC-managed partitions associated with cec01 on cec02, first save the lpars from cec01 to a file:

```
lsvm lpar01-lpar04 > /tmp/myprofile
```

then create lpars on cec02:

```
mkvm lpar05-lpar08 -c cec02 -p /tmp/myprofile
```

Output is similar to:

```
lpar5: Success
lpar6: Success
lpar7: Success
lpar8: Success
```

4. To duplicate all the HMC-managed partitions associated with cec01 on cec02, one is for cec01, the other is for cec02:

```
mkdef -t node -o lpar5,lpar6 mgt=hmc groups=all
chtab node=lpar5 ppc.parent=cec01
chtab node=lpar6 ppc.parent=cec02
```

then create lpars on cec01 and cec02:

```
mkvm lpar5,lpar6 --full
```

Output is similar to:

```
lpar5: Success
lpar6: Success
```

5. To create a new zVM virtual machine (gpok3) based on a directory entry:

```
mkvm gpok3 /tmp/dirEntry.txt
```

Output is similar to:

```
gpok3: Creating user directory entry for LNX3... Done
```

6. To clone a new zVM virtual machine with the same profile/resources as the specified node:

```
mkvm gpok4 gpok3 pool=POOL1
```

Output is similar to:

```
gpok4: Cloning gpok3
gpok4: Linking source disk (0100) as (1100)
gpok4: Linking source disk (0101) as (1101)
gpok4: Stopping LNX3... Done
gpok4: Creating user directory entry
gpok4: Granting VSwitch (VSW1) access for gpok3
gpok4: Granting VSwitch (VSW2) access for gpok3
gpok4: Adding minidisk (0100)
gpok4: Adding minidisk (0101)
gpok4: Disks added (2). Disks in user entry (2)
gpok4: Linking target disk (0100) as (2100)
gpok4: Copying source disk (1100) to target disk (2100) using FLASHCOPY
```

```
gpok4: Mounting /dev/dasdg1 to /mnt/LNX3
gpok4: Setting network configuration
gpok4: Linking target disk (0101) as (2101)
gpok4: Copying source disk (1101) to target disk (2101) using FLASHCOPY
gpok4: Powering on
gpok4: Detatching source disk (0101) at (1101)
gpok4: Detatching source disk (0100) at (1100)
gpok4: Starting LNX3... Done
```

7. To create a new kvm/vmware virtual machine with 10G storage, 2048M memory and 2 cpus.

```
mkvm vm1 -s 10G --mem 2048 --cpus 2
```

8. To create a full partition on normal power machine.

First, define a node object:

```
mkdef -t node -o lpar1 mgt=fsp cons=fsp nodetype=ppc,osi id=1 hcp=cec parent=cec␣
→hwtype=lpar groups=lpar,all
```

Then, create the partion on the specified cec.

```
mkvm lpar1 --full
```

The output is similar to:

```
lpar1: Done
```

To query the resources allocated to node 'lpar1'

```
lsvm lpar1
```

The output is similar to:

```
lpar1: Lpar Processor Info:
Curr Processor Min: 1.
Curr Processor Req: 16.
Curr Processor Max: 16.
lpar1: Lpar Memory Info:
Curr Memory Min: 0.25 GB(1 regions).
Curr Memory Req: 30.75 GB(123 regions).
Curr Memory Max: 32.00 GB(128 regions).
lpar1: 1,519,U78AA.001.WZSGVU7-P1-C7,0x21010207,0xffff(Empty Slot)
lpar1: 1,518,U78AA.001.WZSGVU7-P1-C6,0x21010206,0xffff(Empty Slot)
lpar1: 1,517,U78AA.001.WZSGVU7-P1-C5,0x21010205,0xffff(Empty Slot)
lpar1: 1,516,U78AA.001.WZSGVU7-P1-C4,0x21010204,0xffff(Empty Slot)
lpar1: 1,514,U78AA.001.WZSGVU7-P1-C19,0x21010202,0xffff(Empty Slot)
lpar1: 1,513,U78AA.001.WZSGVU7-P1-T7,0x21010201,0xc03(USB Controller)
lpar1: 1,512,U78AA.001.WZSGVU7-P1-T9,0x21010200,0x104(RAID Controller)
lpar1: 1/2/2
lpar1: 256.
```

Note: The 'parent' attribute for node 'lpar1' is the object name of physical power machine that the full partition will be created on.

9. To create a partition using some of the resources on normal power machine.

Option 1:

After a node object is defined, the resources that will be used for the partition shall be specified like this:

```
chdef lpar1 vmcpus=1/4/16 vmmemory=1G/4G/32G vmphyslots=0x21010201,0x21010200␣
→vmothersetting=bsr:128,hugepage:2
```

Then, create the partion on the specified cec.

```
mkvm lpar1
```

Option 2:

```
mkvm lpar1 vmcpus=1/4/16 vmmemory=1G/4G/32G vmphyslots=0x21010201,0x21010200␣
→vmothersetting=bsr:128,hugepage:2
```

The outout is similar to:

```
lpar1: Done
```

Note: The 'vmplyslots' specify the drc index of the physical slot device. Every drc index shall be delimited with ','. The 'vmothersetting' specify two kinds of resource, bsr(Barrier Synchronization Register) specified the num of BSR arrays, hugepage(Huge Page Memory) specified the num of huge pages.

To query the resources allocated to node 'lpar1'

```
lsvm lpar1
```

The output is similar to:

```
lpar1: Lpar Processor Info:
Curr Processor Min: 1.
Curr Processor Req: 4.
Curr Processor Max: 16.
lpar1: Lpar Memory Info:
Curr Memory Min: 1.00 GB(4 regions).
Curr Memory Req: 4.00 GB(16 regions).
Curr Memory Max: 32.00 GB(128 regions).
lpar1: 1,513,U78AA.001.WZSGVU7-P1-T7,0x21010201,0xc03(USB Controller)
lpar1: 1,512,U78AA.001.WZSGVU7-P1-T9,0x21010200,0x104(RAID Controller)
lpar1: 1/2/2
lpar1: 128.
```

10. To create a vios partition using some of the resources on normal power machine.

```
mkvm viosnode vmcpus=1/4/16 vmmemory=1G/4G/32G vmphyslots=0x21010201,0x21010200␣
→vmnics=vlan1 vmstorage=5 --vios
```

The resouces for the node is similar to:

```
viosnode: Lpar Processor Info:
Curr Processor Min: 1.
Curr Processor Req: 4.
Curr Processor Max: 16.
viosnode: Lpar Memory Info:
Curr Memory Min: 1.00 GB(4 regions).
Curr Memory Req: 4.00 GB(16 regions).
Curr Memory Max: 32.00 GB(128 regions).
viosnode: 1,513,U78AA.001.WZSGVU7-P1-T7,0x21010201,0xc03(USB Controller)
viosnode: 1,512,U78AA.001.WZSGVU7-P1-T9,0x21010200,0x104(RAID Controller)
```

```
viosnode: 1,0,U8205.E6B.0612BAR-V1-C,0x30000000,vSerial Server
viosnode: 1,1,U8205.E6B.0612BAR-V1-C1,0x30000001,vSerial Server
viosnode: 1,3,U8205.E6B.0612BAR-V1-C3,0x30000003,vEth (port_vlanid=1,mac_
→addr=4211509276a7)
viosnode: 1,5,U8205.E6B.0612BAR-V1-C5,0x30000005,vSCSI Server
viosnode: 1,6,U8205.E6B.0612BAR-V1-C6,0x30000006,vSCSI Server
viosnode: 1,7,U8205.E6B.0612BAR-V1-C7,0x30000007,vSCSI Server
viosnode: 1,8,U8205.E6B.0612BAR-V1-C8,0x30000008,vSCSI Server
viosnode: 1,9,U8205.E6B.0612BAR-V1-C9,0x30000009,vSCSI Server
viosnode: 0/0/0
viosnode: 0.
```

FILES

/opt/xcat/bin/mkvm

SEE ALSO

chvm(1)|chvm.1, lsvm(1)|lsvm.1, rmvm(1)|rmvm.1

mkzone.1

NAME

mkzone - Defines a new zone in the cluster.

SYNOPSIS

mkzone <zonename> [**--defaultzone**] [**-k** *full path to the ssh RSA private key*] [**-a** *noderange*] [**-g**] [**-f**] [**-s** *yes|no*] [-V]

mkzone [**-h** | **-v**]

DESCRIPTION

The **mkzone** command is designed to divide the xCAT cluster into multiple zones. The nodes in each zone will share common root ssh keys. This allows the nodes in a zone to be able to as root ssh to each other without password, but cannot do the same to any node in another zone. All zones share a common xCAT Management Node and database including the site table, which defines the attributes of the entire cluster. The mkzone command is only supported on Linux (No AIX support). The nodes are not updated with the new root ssh keys by mkzone. You must run updatenode -k or xdsh -K to the nodes to update the root ssh keys to the new generated zone keys. This will also sync any service nodes with the zone keys, if you have a hierarchical cluster. Note: if any zones in the zone table, there must be one and only one defaultzone. Otherwise, errors will occur.

OPTIONS

-h | --help

> Displays usage information.

-v | --version

> Displays command version and build date.

-k | --sshkeypath *full path to the ssh RSA private key*

> This is the path to the id_rsa key that will be used to build root's ssh keys for the zone. If -k is used, it will generate the ssh public key from the input ssh RSA private key and store both in /etc/xcat/sshkeys/<zonename>/.ssh directory. If -f is not used, then it will generate a set of root ssh keys for the zone and store them in /etc/xcat/sshkeys/<zonename>/.ssh.

--default

> if –defaultzone is input, then it will set the zone defaultzone attribute to yes; otherwise it will set to no. if –defaultzone is input and another zone is currently the default, then the -f flag must be used to force a change to the new defaultzone. If -f flag is not use an error will be returned and no change made. Note: if any zones in the zone table, there must be one and only one defaultzone. Otherwise, errors will occur.

-a | --addnoderange *noderange*

> For each node in the noderange, it will set the zonename attribute for that node to the input zonename. If the -g flag is also on the command, then it will add the group name "zonename" to each node in the noderange.

-s| --sshbetweennodes yes|no

> If -s entered, the zone sshbetweennodes attribute will be set to yes or no. It defaults to yes. When this is set to yes, then ssh will be setup to allow passwordless root access between nodes. If no, then root will be prompted for a password when running ssh between the nodes in the zone.

-f | --force

> Used with the (–defaultzone) flag to override the current default zone.

-g | --assigngroup

> Used with the (-a) flag to create the group zonename for all nodes in the input noderange.

-V | --Verbose

> Verbose mode.

Examples

*

> To make a new zone1 using defaults , enter:
>
> **mkzone** *zone1*
>
> **Note: with the first mkzone, you will automatically get the xcatdefault zone created as the default zone. This zone uses ssh**
> > <roothome>/.ssh directory.

*

> To make a new zone2 using defaults and make it the default zone enter:
>
> **mkzone** *zone2* -****-defaultzone -f

*

> To make a new zone2A using the ssh id_rsa private key in /root/.ssh:
>
> **mkzone** *zone2A* -k /root/.ssh

*

> To make a new zone3 and assign the noderange compute3 to the zone enter:
>
> **mkzone** *zone3* -a compute3

*

> To make a new zone4 and assign the noderange compute4 to the zone and add zone4 as a group to each
> node enter:
>
> **mkzone** *zone4* -a compute4 -g

*

> To make a new zone5 and assign the noderange compute5 to the zone and add zone5 as a group to each
> node but not allow passwordless ssh between the nodes enter:
>
> **mkzone** *zone5* -a compute5 -g -s no

Files

/opt/xcat/bin/mkzone/

Location of the mkzone command.

SEE ALSO

chzone(1)|chzone.1, rmzone(1)|rmzone.1, xdsh(1)|xdsh.1, updatenode(1)|updatenode.1

monadd.1

NAME

monadd - Registers a monitoring plug-in to the xCAT cluster.

SYNOPSIS

monadd [-h| –help]

monadd [-v| –version]

monadd name [-n|–nodestatmon] [-s|–settings settings]

DESCRIPTION

This command is used to register a monitoring plug-in module to monitor the xCAT cluster. The plug-in module will
be added to the xCAT *monitoring* database table and the configuration scripts for the monitoring plug-in, if any, will
be added to the *postscripts* table. A monitoring plug-in module acts as a bridge that connects a 3rd party monitoring
software and the xCAT cluster. A configuration script is used to configure the 3rd party software. Once added to the
<postscripts> table, it will be invoked on the nodes during node deployment stage.

Parameters

name is the name of the monitoring plug-in module. For example, if the the *name* is called *xxx*, then the actual file name that the xcatd looks for is */opt/xcat/lib/perl/xCAT_monitoring/xxx.pm*. Use *monls -a* command to list all the monitoring plug-in modules that can be used.

settings is the monitoring plug-in specific settings. It is used to customize the behavior of the plug-in or configure the 3rd party software. Format: *-s key-value -s key=value ...* Please note that the square brackets are needed here. Use *monls name -d* command to look for the possbile setting keys for a plug-in module.

OPTIONS

-h | --help

> Display usage message.

-n | --nodestatmon

> Indicate that this monitoring plug-in will be used for feeding the node liveness status to the xCAT *nodelist* table.

-s | --settings

> Specifies the plug-in specific settings. These settings will be used by the plug-in to customize certain entities for the plug-in or the third party monitoring software. e.g. -s mon_interval=10 -s toggle=1.

-v | --version ****

> Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To register gangliamon plug-in module (which interacts with Ganglia monitoring software) to monitor the xCAT cluster, enter:

```
monadd gangliamon
```

2. To register rmcmon plug-in module (which interacts with IBM's RSCT monitoring software) to monitor the xCAT cluster and have it feed the node liveness status to xCAT's *nodelist* table, enter:

```
monadd rmcmon -n
```

This will also add the *configrmcnode* to the *postscripts* table. To view the content of the *postscripts* table, enter:

```
tabdump postscritps
#node,postscripts,comments,disable
"service","servicenode",,
"xcatdefaults","syslog,remoteshell,configrmcnode",,
```

3. To register xcatmon plug-in module to feed the node liveness status to xCAT's *nodelist* table, enter:

```
monadd xcatmon -n -s ping-interval=2
```

where 2 is the number of minutes between the pings.

FILES

/opt/xcat/bin/monadd

SEE ALSO

monls(1)|monls.1, monrm(1)|monrm.1, monstart(1)|monstart.1, monstop(1)|monstop.1, moncfg(1)|moncfg.1, mondecfg(1)|mondecfg.1

moncfg.1

NAME

moncfg - Configures a 3rd party monitoring software to monitor the xCAT cluster.

SYNOPSIS

moncfg [-h| –help]

moncfg [-v| –version]

moncfg name [noderange] [-r|–remote]

DESCRIPTION

This command is used to configure a 3rd party monitoring software to monitor the xCAT cluster. For example, it modifies the configuration file for the monitoring software so that the nodes can be included in the monitoring domain. The operation is performed on the management node and the service nodes of the given nodes. The operation will also be performed on the nodes if the *-r* option is specified, though the configuration of the nodes is usually performed during the node deployment stage.

Parameters

name is the name of the monitoring plug-in module. For example, if the the *name* is called *xxx*, then the actual file name that the xcatd looks for is */opt/xcat/lib/perl/xCAT_monitoring/xxx.pm*. Use *monls -a* command to list all the monitoring plug-in modules that can be used.

noderange specifies the nodes to be monitored. If omitted, all nodes will be monitored.

OPTIONS

-h | --help Display usage message.

-r | --remote Specifies that the operation will also be performed on the nodes.

-v | -****-version ****** Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To configure the management node and the service nodes for ganglia monitoring, enter:

```
moncfg gangliamon
```

1. To configure the management node, nodes and their service nodes for ganglia monitoring, enter:

```
moncfg gangliamon -r
```

FILES

/opt/xcat/bin/moncfg

SEE ALSO

monls(1)|monls.1, mondecfg(1)|mondecfg.1, monadd(1)|monadd.1, monrm(1)|monrm.1, monstart(1)|monstart.1, monstop(1)|monstop.1

mondecfg.1

NAME

mondecfg - Deconfigures a 3rd party monitoring software from monitoring the xCAT cluster.

SYNOPSIS

moncfg [-h| –help]

moncfg [-v| –version]

moncfg name [noderange] [-r|–remote]

DESCRIPTION

This command is used to deconfigure a 3rd party monitoring software from monitoring the xCAT cluster. The operation is performed on the management node and the service nodes of the given nodes. The operation will also be performed on the nodes if the *-r* option is specified. The deconfigration operation will remove the nodes from the 3rd party software's monitoring domain.

Parameters

name is the name of the monitoring plug-in module. Use *monls* command to list all the monitoring plug-in modules that can be used.

noderange specified the nodes to be deconfigured. If omitted, all nodes will be deconfigured.

OPTIONS

-h | --help Display usage message.

-r | --remote Specifies that the operation will also be performed on the nodes.

-v | --version** Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To deconfigure the management node and the service nodes from the ganglia monitoring, enter:

```
mondecfg gangliamon
```

1. To deconfigure the management node, nodes and their service nodes from the ganglia monitoring, enter:

```
mondecfg gangliamon -r
```

FILES

/opt/xcat/bin/mondecfg

SEE ALSO

monls(1)|monls.1, moncfg(1)|moncfg.1, monadd(1)|monadd.1, monrm(1)|monrm.1, monstart(1)|monstart.1, monstop(1)|monstop.1

monls.1

NAME

monls - Lists monitoring plug-in modules that can be used to monitor the xCAT cluster.

SYNOPSIS

monls [-h| –help]

monls [-v| –version]

monls *name* [-d|–description]

monls [-a|–all] [-d|–description]

DESCRIPTION

This command is used to list the status, desctiption, the configuration scripts and the settings of one or all of the monitoring plug-in modules.

Parameters

name is the name of the monitoring plug-in module.

OPTIONS

-a | --all Searches the *XCATROOT/lib/perl/xCAT_monitoring* directory and reports all the monitoring plug-in modules. If nothing is specified, the list is read from the *monitoring* tabel.

-d | --description Display the description of the plug-in modules. The description ususally contains the possible settings.

-h | --help Display usage message.

-v | --version** Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To list the status of all the monitoring plug-in modules from the *monitoring* table, enter:

```
monls
```

The output looks like this:

```
xcatmon          monitored          node-status-monitored
snmpmon          not-monitored
```

2. To list the status of all the monitoring plug-in modules including the ones that are not in the monitoring table, enter

```
monls -a
```

The output looks like this:

```
xcatmon          monitored          node-status-monitored
snmpmon          not-monitored
gangliamon       not-monitored
rmcmon           monitored
nagiosmon        not-monitored
```

3. To list the status and the desciption for *snmpmon* module, enter:

```
monls snmpmon -d
```

FILES

/opt/xcat/bin/monls

SEE ALSO

monadd(1)|monadd.1, monrm(1)|monrm.1, monstart(1)|monstart.1, monstop(1)|monstop.1, moncfg(1)|moncfg.1, mondecfg(1)|mondecfg.1

monrm.1

NAME

monrm - Unregisters a monitoring plug-in module from the xCAT cluster.

SYNOPSIS

monrm [-h| –help]

monrm [-v| –version]

monrm name

DESCRIPTION

This command is used to unregister a monitoring plug-in module from the *monitoring* table. It also removes any configuration scripts associated with the monitoring plug-in from the *postscripts* table. A monitoring plug-in module acts as a bridge that connects a 3rd party monitoring software and the xCAT cluster. A configuration script is used to configure the 3rd party software. Once added to the *postscripts* table, it will be invoked on the nodes during node deployment stage.

Parameters

name is the name of the monitoring plug-in module in the *monitoring* table. Use *monls* command to list all the monitoring plug-in modules that can be used.

OPTIONS

-h | --help Display usage message.

-v | -**-version ****** Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1.To unregister gangliamon plug-in module (which interacts with Ganglia monitoring software) from the xCAT cluster, enter:

```
monrm gangliamon
```

Please note that gangliamon must have been registered in the xCAT *monitoring* table. For a list of registered plug-in modules, use command *monls*.

FILES

/opt/xcat/bin/monrm

SEE ALSO

monls(1)|monls.1, monadd(1)|monadd.1, monstart(1)|monstart.1, monstop(1)|monstop.1, moncfg(1)|moncfg.1, mondecfg(1)|mondecfg.1

monshow.1

NAME

monshow - Shows event data for monitoring.

SYNOPSIS

monshow [-h| –help]

monshow [-v| –version]

monshow name [noderange] [-s] [-t time] [-a attributes] [-w attr<operator>val [-w attr<operator>val] ...][-o {p|e}]>

DESCRIPTION

This command displays the events that happened on the given nodes or the monitoring data that is collected from the given nodes for a monitoring plugin.

Parameters

name is the name of the monitoring plug-in module to be invoked.

noderange is a list of nodes to be showed for. If omitted, the data for all the nodes will be displayed.

OPTIONS

-h | --help Display usage message.

-v | --version ** Command Version.

-s shows the summary data.

-t specifies a range of time for the data, The default is last 60 minutes. For example -t 6-4, it will display the data from last 6 minutes to 4 minutes; If it is -t 6, it will display the data from last 6 minutes until now.

-a specifies a comma-separated list of attributes or metrics names. The default is all.

-w specify one or multiple selection string that can be used to select events. The operators ==, !=, =~,!,>,<,>=,<= are available. Wildcards % and _ are supported in the pattern string. % allows you to match any string of any length(including zero length) and _ allows you to match on a single character. The valid attributes are eventtype, monitor, monnode, application, component, id, serverity, message, rawdata, comments. Valid severity are: Informational, Warning, Critical.

Operator descriptions: == Select event where the attribute value is exactly this value. != Select event where the attribute value is not this specific value. =~ Select event where the attribute value matches this pattern string. Not work with severity. !~ Select event where the attribute value does not match this pattern string. Not work with severity. > Select event where the severity is higher than this value. Only work with severity. < Select event where the severity is lower than this value. Only work with severity. >= Select event where the severity is higher than this value(include). Only work with severity. <= Select event where the severity is lower than this value(include). Only work with severity.

> Note: if the "val" or "operator" fields includes spaces or any other characters that will be parsed by shell, the "attr<operator>val" needs to be quoted. If the operator is "!~", the "attr<operator>val" needs to be quoted using single quote.

-o specifies montype, it can be p or e. p means performance, e means events.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To show summary data about PctRealMemFree and PctTotalTimeIdle of cluster in last 6 minutes, enter:

```
monshow rmcmon -s -a PctRealMemFree,PctTotalTimeIdle -t 6
```

2. To show all data of node1 and node2, enter:

```
monshow rmcmon node1,node2
```

3. To show summary data of nodes which managed by servicenode1, enter:

```
monshow rmcmon servicenode1 -s
```

4. To show RMC event with severity Critical, enter:

```
monshow rmcmon -w severity==Critical
```

FILES

/opt/xcat/bin/monshow

SEE ALSO

monls(1)|monls.1, monstart(1)|monstart.1, monstop(1)|monstop.1, monadd(1)|monadd.1, monrm(1)|monrm.1, moncfg(1)|moncfg.1, mondecfg(1)|mondecfg.1

monstart.1

NAME

monstart - Starts a plug-in module to monitor the xCAT cluster.

SYNOPSIS

monstart [-h| –help]

monstart [-v| –version]

monstart name [noderange] [-r|–remote]

DESCRIPTION

This command is used to start a 3rd party software, (for example start the daemons), to monitor the xCAT cluster. The operation is performed on the management node and the service nodes of the given nodes. The operation will also be performed on the nodes if the *-r* option is specified.

Parameters

name is the name of the monitoring plug-in module. For example, if the the *name* is called *xxx*, then the actual file name that the xcatd looks for is */opt/xcat/lib/perl/xCAT_monitoring/xxx.pm*. Use *monls -a* command to list all the monitoring plug-in modules that can be used.

noderange is the nodes to be monitored. If omitted, all nodes will be monitored.

OPTIONS

-h | --help Display usage message.

-r | --remote Specifies that the operation will also be performed on the nodes. For example, the3rd party monitoring software daemons on the nodes will also be started.

-v | --version ** Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To start gangliamon plug-in module (which interacts with Ganglia monitoring software) to monitor the xCAT cluster, enter:

```
monstart gangliamon -r
```

2. To start xcatmon plug-in module to feed the node liveness status to xCAT's *nodelist* table, enter:

```
monstart rmcmon
```

FILES

/opt/xcat/bin/monstart

SEE ALSO

monls(1)|monls.1, monstop(1)|monstop.1, monadd(1)|monadd.1, monrm(1)|monrm.1, moncfg(1)|moncfg.1, mondecfg(1)|mondecfg.1

monstop.1

NAME

monstop - Stops a monitoring plug-in module to monitor the xCAT cluster.

SYNOPSIS

monstop [-h| –help]

monstop [-v| –version]

monstop name [noderange] [-r|–remote]

DESCRIPTION

This command is used to stop a 3rd party software, (for example stop the daemons), from monitoring the xCAT cluster. The operation is performed on the management node and the service nodes of the given nodes. The operation will also be performed on the nodes if the *-r* option is specified.

Parameters

name is the name of the monitoring plug-in module in the *monitoring* table. Use *monls* command to list all the monitoring plug-in modules that can be used.

noderange is the nodes to be stopped for monitoring. If omitted, all nodes will be stopped.

OPTIONS

-h | -help Display usage message.

-r | --remote Specifies that the operation will also be performed on the nodes. For example, the3rd party monitoring software daemons on the nodes will also be stopped.

-v | -version **** Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1.To stop gangliamon plug-in module (which interacts with Ganglia monitoring software) to monitor the xCAT cluster, enter:

```
monstop gangliamon
```

Please note that gangliamon must have been registered in the xCAT *monitoring* table. For a list of registered plug-in modules, use command *monls*.

FILES

/opt/xcat/bin/monstop

SEE ALSO

monls(1)|monls.1, monstart(1)|monstart.1, monadd(1)|monadd.1, monrm(1)|monrm.1, moncfg(1)|moncfg.1, mondecfg(1)|mondecfg.1

mysqlsetup.1

NAME

mysqlsetup - Sets up the MySQL or MariaDB database for xCAT to use.

SYNOPSIS

mysqlsetup {**-h | --help**}

mysqlsetup {**-v | --version**}

mysqlsetup {**-i | --init**} [**-f | --hostfile**] [-o|-****-odbc]** [**-L|--LL**] [****-V | --verbose**]

mysqlsetup {**-u | --update**} [**-f | --hostfile**] [-o|-****-odbc]** [**-L|--LL**] [****-V | --verbose**]

mysqlsetup {**-o | --odbc**} [-V|-********-verbose]

mysqlsetup {**-L | --LL**} [-V|-********-verbose]

DESCRIPTION

mysqlsetup - Sets up the MySQL or MariaDB database (linux only for MariaDB) for xCAT to use. The mysqlsetup script is run on the Management Node as root after the MySQL code or MariaDB code has been installed. Before running the init option, the MySQL server should be stopped, if it is running. The xCAT daemon, xcatd, must be running, do not stop it. No xCAT commands should be run during the init process, because we will be migrating the xCAT database to MySQL or MariaDB and restarting the xcatd daemon as well as the MySQL daemon. For full information on all the steps that will be done, read the "Configure MySQL and Migrate xCAT Data to MySQL" sections in Setting_Up_MySQL_as_the_xCAT_DB Two passwords must be supplied for the setup, a password for the xcatadmin id and a password for the root id in the MySQL database. These will be prompted for interactively, unless the environment variables XCATMYSQLADMIN_PW and XCATMYSQLROOT_PW are set to the passwords for the xcatadmin id and root id in the database,resp. Note below we refer to MySQL but it works the same for MariaDB.

OPTIONS

-h|--help

> Displays the usage message.

-v|--version

> Displays the release version of the code.

-V|--verbose

> Displays verbose messages.

-i|--init

> The init option is used to setup a xCAT database on an installed MySQL or MariaDB server for xCAT to use. The mysqlsetup script will check for the installed MariaDB server rpm first and will use MariaDB if it is installed. This involves creating the xcatdb database, the xcatadmin id, allowing access to the xcatdb database by the Management Node. It customizes the my.cnf configuration file for xcat and starts the MySQL server. It also backs up the current xCAT database and restores it into the newly setup xcatdb MySQL database. It creates the /etc/xcat/cfgloc file to point the xcatd daemon to the MySQL database and restarts the xcatd daemon using the database. On AIX, it additionally setup the mysql id and group

and corrects the permissions in the MySQL install directories. For AIX, you should be using the MySQL rpms available from the xCAT website. For Linux, you should use the MySQL or MariaDB rpms shipped with the OS. You can chose the -f and/or the -o option, to run after the init.

-u|--update

To run the update option, you must first have run the -i option and have xcat successfully running on the MySQL database. You can chose the -f and/or the -o option, to update.

-f|--hostfile

This option runs during update, it will take all the host from the input file (provide a full path) and give them database access to the xcatdb in MySQL for the xcatadmin id. Wildcards and ipaddresses may be used. xCAT must have been previously successfully setup to use MySQL. xcatadmin and MySQL root password are required.

-o|--odbc

This option sets up the ODBC /etc/../odbcinst.ini, /etc/../odbc.ini and the .odbc.ini file in roots home directory will be created and initialized to run off the xcatdb MySQL database. See "Add ODBC Support" in Setting_Up_MySQL_as_the_xCAT_DB

-L|--LL

Additional database configuration specifically for the LoadLeveler product. See "Add ODBC Support" in Setting_Up_MySQL_as_the_xCAT_DB

ENVIRONMENT VARIABLES

*

XCATMYSQLADMIN_PW - the password for the xcatadmin id that will be assigned in the MySQL database.

*

XCATMYSQLROOT_PW - the password for the root id that will be assigned to the MySQL root id, if the script creates it. The password to use to run MySQL command to the database as the MySQL root id. This password may be different than the unix root password on the Management Node.

EXAMPLES

*

To setup MySQL for xCAT to run on the MySQL xcatdb database :

mysqlsetup *-i*

*

Add hosts from /tmp/xcat/hostlist that can access the xcatdb database in MySQL:

mysqlsetup *-u -f /tmp/xcat/hostlist*

Where the file contains a host per line, for example:

```
node1
1.115.85.2
10.%.%.%
nodex.cluster.net
```

*

> To setup the ODBC for MySQL xcatdb database access :

> **mysqlsetup** *-o*

*

> To setup MySQL for xCAT and add hosts from /tmp/xcat/hostlist and setup the ODBC in Verbose mode:

> **mysqlsetup** *-i -f /tmp/xcat/hostlist -o -V*

nimnodecust.1

NAME

nimnodecust - Use this xCAT command to customize AIX/NIM standalone machines.

SYNOPSIS

nimnodecust [-h|--help]

nimnodecust [-V] -s lpp_source_name [-p packages] [-b installp_bundles] noderange [attr=val [attr=val ...]]

DESCRIPTION

This xCAT command can be used to customize AIX/NIM standalone machines.

The software packages that you wish to install on the nodes must be copied to the appropriate directory locations in the NIM lpp_source resource provided by the "-s" option. For example, if the location of your lpp_source resource is "/install/nim/lpp_source/61lpp/" then you would copy RPM packages to "/install/nim/lpp_source/61lpp/RPMS/ppc" and you would copy your installp packages to "/install/nim/lpp_source/61lpp/installp/ppc". Typically you would want to copy the packages to the same lpp_source that was used to install the node. You can find the location for an lpp_source with the AIX lsnim command. (Ex. "lsnim -l <lpp_source_name>")

The packages you wish to install on the nodes may be specified with either a comma-separated list of package names or by a comma-separated list of installp_bundle names. The installp_bundle names are what were used when creating the corresponding NIM installp_bundle definitions. The installp_bundle definitions may also be used when installing the nodes.

A bundle file contains a list of package names. The RPMs must have a prefix of "R:" and the installp packages must have a prefix of "I:". For example, the contents of a simple bundle file might look like the following.

```
# RPM
R:expect-5.42.1-3.aix5.1.ppc.rpm
R:ping-2.4b2_to-1.aix5.3.ppc.rpm

#installp
I:openssh.base
I:openssh.license
```

To create a NIM installp_bundle definition you can use the "nim -o define" operation. For example, to create a definition called "mypackages" for a bundle file located at "/install/nim/mypkgs.bnd" you could issue the following command.

"nim -o define -t installp_bundle -a server=master -a location=/install/nim/mypkgs.bnd mypackages".

See the AIX documantation for more information on using installp_bundle files.

The xCAT nimnodecust command will automatically handle the distribution of the packages to AIX service nodes when using an xCAT hierachical environment.

OPTIONS

attr=val [attr=val ...]

> Specifies one or more "attribute equals value" pairs, separated by spaces. Attr=val pairs must be specified last on the command line. These are used to specify additional values that can be passed to the underlying NIM commands, ("nim -o cust..."). See the NIM documentation for valid "nim" command line options.

-b installp_bundle_names

```
A comma separated list of NIM installp_bundle names.
```

-h |--help

> Display usage message.

-p package_names

> A comma-separated list of software packages to install. Packages may be RPM or installp.

noderange

> A set of comma delimited node names and/or group names. See the "noderange" man page for details on additional supported formats.

-V |--verbose

> Verbose mode.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

1. Install the installp package "openssh.base.server" on an xCAT node named "node01". Assume that the package has been copied to the NIM lpp_source resource called "61lppsource".

nimnodecust -s 61lppsource -p openssh.base.server node01

2. Install the product software contained in the two bundles called "llbnd" and "pebnd" on all AIX nodes contained in the xCAT node group called "aixnodes". Assume that all the software packages have been copied to the NIM lpp_source resource called "61lppsource".

nimnodecust -s 61lppsource -b llbnd,pebnd aixnodes

FILES

/opt/xcat/bin/nimnodecust

NOTES

This command is part of the xCAT software product.

nimnodeset.1

NAME

nimnodeset - Use this xCAT command to initialize AIX/NIM standalone machines.

SYNOPSIS

nimnodeset [-h|--help]

nimnodeset [-V|--verbose] [-f|--force] [-i osimage_name] [-l location] [-p|--primarySN] [-b|--backupSN] noderange [attr=val [attr=val ...]]

DESCRIPTION

This xCAT command can be used to initialize AIX/NIM standalone machines. Once this step is completed the either the xCAT **rnetboot** command or the **rbootseq/rpower** commands to initiate a network boot of the nodes.

If you are using xCAT service nodes the **nimnodeset** command will automatically determine the correct server(s) for the node and do the initialization on that server(s).

The osimage_name is the name of an xCAT osimage definition that contains the list of NIM resources to use when initializing the nodes. If the osimage_name is not provided on the command line the code checks the node definition for the value of the "provmethod" attribute (which is the name of an osimage definition). If the osimage_image is provided on the command line then the code will also set the "provmethod" attribute of the node definiions.

This command will also create a NIM resolv_conf resource to be used when installing the node. If a resolv_conf resource is not already included in the xCAT osimage definition and if the "domain" and "nameservers" values are set then a new NIM resolv_conf resource will be created and allocated to the nodes.

The "domain" and "nameservers" attributes can be set in either the xCAT "network" definition used by the nodes or in the xCAT cluster "site" definition. The setting in the "network" definition will take priority.

The "search" field of the resolv.conf file will contain a list all the domains listed in the xCAT network definitions and the xCAT site definiton.

The "nameservers" value can either be set to a specific IP address or the "<xcatmaster>" key word. The "<xcatmaster>" key word means that the value of the "xcatmaster" attribute of the node definition will be used in the /etc/resolv.conf file. (I.e. The name of the install server as known by the node.)

You can set the "domain" and "nameservers" attributes by using the **chdef** command. For example:

chdef -t network -o clstr_net domain=cluster.com nameservers=<xcatmaster>

If the "domain" and "nameservers" attributes are not set in either the nodes "network" definition or the "site" definition then no new NIM resolv_conf resource will be created.

You can specify additional attributes and values using the "attr=val" command line option. This information will be passed on to the underlying call to the NIM "nim -o bos_inst" command. See the NIM documentation for information on valid command line options for the nim command. The "attr" must correspond to a NIM attribute supported for the NIM "bos_inst" operation. Information provided by the "attr=val" option will take precedence over the information provided in the osimage definition.

The force option can be used to reinitialize a node if it already has resources allocated or it is in the wrong NIM state. This option will reset the NIM node and deallocate resources before reinititializing.

This command will also create a NIM script resource to enable the xCAT support for user-provided customization scripts.

After the **nimnodeset** command completes you can use the **lsnim** command to check the NIM node definition to see if it is ready for booting the node. ("lsnim -l <nim_node_name>").

You can supply your own scripts to be run on the management node or on the service node (if their is hierarchy) for a node during the **nimnodeset** command. Such scripts are called **prescripts**. They should be copied to /install/prescripts dirctory. A table called *prescripts* is used to specify the scripts and their associated actions. The scripts to be run at the beginning of the **nimnodeset** command are stored in the 'begin' column of *prescripts* table. The scripts to be run at the end of the **nimnodeset** command are stored in the 'end' column of *prescripts* table. Please run 'tabdump prescripts -d' command for details. An example for the 'begin' or the 'end' column is: *standalone:myscript1,myscript2*. The following two environment variables will be passed to each script: NODES contains all the names of the nodes that need to run the script for and ACTION contains the current nodeset action, in this case "standalone". 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 will be invoked at at a time. If it is not specified, the script will be invoked once for all the nodes.

OPTIONS

attr=val [attr=val ...]

> Specifies one or more "attribute equals value" pairs, separated by spaces. Attr= val pairs must be specified last on the command line. These are used to specify additional values that can be passed to the underlying NIM commands, ("nim -o bos_inst ..."). See the NIM documentation for valid "nim" command line options. Note that you may specify multiple "script" and "installp_bundle" values by using a comma seperated list. (ex. "script=ascript,bscript").

-b|--backupSN

> When using backup service nodes only update the backup. The default is to update both the primary and backup service nodes

-f |--force

> Use the force option to reinitialize the NIM machines.

-h |--help

> Display usage message.

-i image_name

> The name of an existing xCAT osimage definition.

-l|--location

> The directory location to use when creating new NIM resolv_conf resources. The d efault location is /install/nim.

-p|--primarySN

When using backup service nodes only update the primary. The default is to update both the primary and backup service nodes.

noderange

A set of comma delimited node names and/or group names. See the "noderange" man page for details on additional supported formats.

-V |--verbose

Verbose mode.

RETURN VALUE

0

The command completed successfully.

1

An error has occurred.

EXAMPLES

1. Initialize an xCAT node named "node01". Use the xCAT osimage named "61gold" to install the node.

nimnodeset -i 61gold node01

2. Initialize all AIX nodes contained in the xCAT node group called "aixnodes" using the image definitions pointed to by the "provmethod" attribute of the xCAT node definitions.

nimnodeset aixnodes

3. Initialize an xCAT node called "node02". Include installp_bundle resources that are not included in the osimage definition. This assumes the NIM installp_bundle resources have already been created.

nimnodeset -i 611image node02 installp_bundle=sshbundle,addswbundle

FILES

/opt/xcat/bin/nimnodeset

NOTES

This command is part of the xCAT software product.

SEE ALSO

mknimimage(1)|mknimimage.1, rnetboot(1)|rnetboot.1

nodeaddunmged.1

NAME

nodeaddunmged - Create a unmanaged node.

SYNOPSIS

nodeaddunmged [-h| -****-help | -v | -****-version]

nodeaddunmged hostname=<node-name> ip=<ip-address>

DESCRIPTION

The **nodeaddunmged** command adds an unmanaged node to the __Unmanaged group. You can specify the node name and IP address of the node.

OPTIONS

-h|--help

Display usage message.

-v|--version

Command Version.

hostname=<node-name>

Sets the name of the new unmanaged node, where <node-name> is the name of the node.

ip=<ip-address>

Sets the IP address of the unmanaged node, where <ip-address> is the IP address of the new node in the form xxx.xxx.xxx.xxx

RETURN VALUE

0 The command completed successfully.

1 An error has occured.

EXAMPLES

To add an unmanaged node, use the following command: nodeaddunmged hostname=unmanaged01 ip=192.168.1.100

SEE ALSO

nodech.1

NAME

nodech - Changes nodes' attributes in the xCAT cluster database.

SYNOPSIS

nodech *noderange table.column=value* [...]

nodech {**-d** | **--delete**} *noderange table* [...]

nodech {**-v** | **--version**}

nodech [**-?** | **-h** | **--help**]

DESCRIPTION

The nodech command changes the specified attributes for the given nodes. Normally, the given value will completely replace the current attribute value. But if ",=" is used instead of "=", the specified value will be prepended to the attribute's comma separated list, if it is not already there. If "^=" is used, the specified value will be removed from the attribute's comma separated list, if it is there. You can also use "^=" and ",=" in the same command to essentially replace one item in the list with another. (See the Examples section.)

Additionally, as in nodels, boolean expressions can be used to further limit the scope of nodech from the given noderange. The operators supported are the same as nodels (=~, !~, ==, and !=).

With these operators in mind, the unambiguous assignment operator is '=@'. If you need, for example, to set the nodelist.comments to =foo, you would have to do *nodech n1 nodelist.comments=@=foo*.

See the **xcatdb** man page for an overview of each table.

The nodech command also supports some short cut names as aliases to common attributes. See the **nodels** man page for details.

OPTIONS

-d|--delete

> Delete the nodes' row in the specified tables.

-v|--version

> Command Version.

-?|-h|--help

> Display usage message.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

*

> To update nodes in noderange node1-node4 to be in only group all:
>
> **nodech** *node1-node4 groups=all*

*

> To put all nodes with nodepos.rack value of 2 into a group called rack2:
>
> **nodech** *all* nodepos.rack==2 groups,=rack2

*

> To add nodes in noderange node1-node4 to the nodetype table with os=rhel5:
>
> **nodech** *node1-node4 groups=all,rhel5 nodetype.os=rhel5*

*

> To add node1-node4 to group1 in addition to the groups they are already in:
>
> **nodech** *node1-node4 groups,=group1*

*

> To put node1-node4 in group2, instead of group1:
>
> **nodech** *node1-node4 groups^=group1 groups,=group2*

FILES

/opt/xcat/bin/nodech

SEE ALSO

nodels(1)|nodels.1, nodeadd(8)|nodeadd.8, noderange(3)|noderange.3

nodechmac.1

NAME

nodechmac - Updates the MAC address for a node.

SYNOPSIS

nodechmac [-h| -****-help** | **-v** | **-****-version]

nodechmac <node-name> mac=<mac-address>

DESCRIPTION

The **nodechmac** command changes the MAC address for provisioned node's network interface.

You can use this command to keep an existing node configuration. For example, if an existing node has hardware problems, the replacement node can use the old configurations. By using the nodechmac command, the node name and network settings of the old node can be used by the new node.

OPTIONS

-h|--help

Display usage message.

-v|--version

Command Version.

node-name

Specifies the name of the node you want to update, where <node-name> is the node that is updated.

mac=<mac-address

Sets the new MAC address for the NIC used by the provisioning node, where <mac-address> is the NICs new MAC address.

RETURN VALUE

0 The command completed successfully.

1 An error has occured.

EXAMPLES

You can update the MAC address for a node, by using the following command: nodechmac compute-000 mac=2F:3C:88:98:7E:01

SEE ALSO

nodechprofile.1

NAME

nodechprofile - updates a profile used by a node

SYNOPSIS

nodechprofile [-h| -****-help | -v | -****-version]

nodechprofile   <noderange>   [imageprofile=<image-profile>]   [networkprofile=<network-profile>] [hardwareprofile=<hardware-profile>]

DESCRIPTION

The **nodechprofile** command updates the profiles used by a node, including: the image profile, network profile, and hardware management profile.

If you update the image profile for a node, the operating system and provisioning settings for the node are updated.

If you update the network profile, the IP address and network settings for the node are updated.

If you update the hardware management profile, the hardware settings for the node are updated.

After nodes' hardware profile or image profile are updated, the status for each node is changed to "defined". A node with a "defined" status must be reinstalled

After nodes' network profile updated, the status for nodes is not changed. You'll need to run **noderegenips** to regenerate the nodes' IP address and nodes' status may also be updated at this stage.

OPTIONS

-h|--help

Display usage message.

-v|--version

Command Version.

noderange

The nodes to be removed.

imageprofile=<image-profile>

Sets the new image profile name used by the node, where <image-profile> is the new image profile. An image profile defines the provisioning method, OS information, kit information, and provisioning parameters for a node. If the "__ImageProfile_imgprofile" group already exists in the nodehm table, then "imgprofile" is used as the image profile name.

networkprofile=<network-profile>

Sets the new network profile name used by the node, where <network-profile> is the new network profile. A network profile defines the network, NIC, and routes for a node. If the "__NetworkProfile_netprofile" group already exists in the nodehm table, then "netprofile" is used as the network profile name.

hardwareprofile=<hardware-profile>

Sets the new hardware profile name used by the node, where <hardware-profile> is the new hardware management profile used by the node. If a "__HardwareProfile_hwprofile" group exists, then "hwprofile" is the hardware profile name. A hardware profile defines hardware management related information for imported nodes, including: IPMI, HMC, CEC, CMM.

RETURN VALUE

0 The command completed successfully.

1 An error has occured.

EXAMPLES

To change the image profile to rhels6.3_packaged for compute nodes compute-000 and compute-001, use the following command:

nodechprofile compute-000,compute-001 imageprofile=rhels6.3_packaged

To change all of the profiles for compute node compute-000, enter the following command:

nodechprofile   compute-000   imageprofile=rhels6.3_packaged   networkprofile=default_cn   hardwareprofile=default_ipmi

SEE ALSO

nodepurge(1)|nodepurge.1, noderefresh(1)|noderefresh.1, nodeimport(1)|nodeimport.1, noderange(3)|noderange.3

nodediscoverdef.1

NAME

nodediscoverdef - Define the undefined discovery request to a predefined xCAT node, or clean up the discovery entries from the discoverydata table (which can be displayed by nodediscoverls command)

SYNOPSIS

nodediscoverdef -u uuid -n node

nodediscoverdef -r -u uuid

nodediscoverdef -r -t {**seq | profile | switch | blade | manual | undef | all**}

nodediscoverdef [**-h | --help | -v | --version**]

DESCRIPTION

The **nodediscoverdef** command defines the discovery entry from the discoverydata table to a predefined xCAT node. The discovery entry can be displayed by **nodediscoverls** command.

The options **-u** and **-n** have to be used together to define a discovery request to a node.

The **nodediscoverdef** command also can be used to clean up the discovery entries from the discoverydata table.

The option **-r** is used to remove discovery entries. If working with **-u**, the specific entry which uuid specified by **-u** will be removed.

You also can use the **-r -t** option to limit that only remove the nodes that were discovered in a particular method of discovery.

OPTIONS

-t seq|profile|switch|blade|manual|undef|all

Specify the nodes that have been discovered by the specified discovery method:

*

>   **seq** - Sequential discovery (started via nodediscoverstart noderange=<noderange> ...).

*

>   **profile** - Profile discovery (started via nodediscoverstart networkprofile=<network-profile> ...).

*

>   **switch** - Switch-based discovery (used when the switch and switches tables are filled in).

*

>   **blade** - Blade discovery (used for IBM Flex blades).

*

>   **manual** - Manually discovery (used when defining node by nodediscoverdef command).

*

>   **undef** - Display the nodes that were in the discovery pool, but for which xCAT has not yet received a discovery request.

*

>   **all** - All discovered nodes.

-n node

>   The xCAT node that the discovery entry will be defined to.

-r

>   Remove the discovery entries from discoverydata table.

-u uuid

>   The uuid of the discovered entry.

-h|--help

>   Display usage message.

-v|--version

>   Command version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1

>   Define the discovery entry which uuid is 51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB4 to node node1
>
>   **nodediscoverdef** -u 51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB4 -n node1

```
Defined [51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB4] to node node1.
```

2

Remove the discovery entry which uuid is 51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB4 from the discoverydata table

nodediscoverdef -r -u 51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB4

```
Removing discovery entries finished.
```

3

Remove the discovery entries which discover type is **seq** from the discoverydata table

nodediscoverdef -r -t seq

```
Removing discovery entries finished.
```

SEE ALSO

nodediscoverstart(1)|nodediscoverstart.1,   nodediscoverstatus(1)|nodediscoverstatus.1,   nodediscoverstop(1)|nodediscoverstop.1, nodediscoverls(1)|nodediscoverls.1

nodediscoverls.1

NAME

nodediscoverls - List the discovered nodes

SYNOPSIS

nodediscoverls [**-t seq | profile | switch | blade | manual | undef | all**] [**-l**]

nodediscoverls [**-u uuid**] [**-l**]

nodediscoverls [**-h | --help | -v | --version**]

DESCRIPTION

The **nodediscoverls** command lists nodes that have recently been discovered. If discovery is currently in progress (i.e. **nodediscoverstart** has been run, but **nodediscoverstop** has not been), then **nodediscoverls** will list the nodes that have been discovered so far in this session. If discovery is not currently in progress, **nodediscoverls** will list all of the nodes that were discovered in the last discovery session.

You can use the **-t** option to limit the output to just the nodes that were discovered in a particular method of discovery.

OPTIONS

-t seq|profile|switch|blade|manual|undef|all

Display the nodes that have been discovered by the specified discovery method:

*

> **seq** - Sequential discovery (started via nodediscoverstart noderange=<noderange> ...).

*

> **profile** - Profile discovery (started via nodediscoverstart networkprofile=<network-profile> ...).

*

> **switch** - Switch-based discovery (used when the switch and switches tables are filled in).

*

> **blade** - Blade discovery (used for IBM Flex blades).

*

> **manual** - Manually discovery (used when defining node by nodediscoverdef command).

*

> **undef** - Display the nodes that were in the discovery pool, but for which xCAT has not yet received a discovery request.

*

> **all** - All discovered nodes.

-l

> Display more detailed information about the discovered nodes.

-u uuid

> Display the discovered node that has this uuid.

-h|--help

> Display usage message.

-v|--version

> Command version.

RETURN VALUE

0 The command completed successfully.

1 An error has occured.

EXAMPLES

1

> Display the discovered nodes when sequential discovery is running:
>
> **nodediscoverls**

```
UUID                                          NODE              METHOD        ␣
↪MTM       SERIAL
51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB2  distest1          sequential    ␣
↪786310X   1052EF2
51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB3  distest2          sequential    ␣
↪786310X   1052EF3
```

2

Display the nodes that were in the discovery pool, but for which xCAT has not yet received a discovery request:

nodediscoverls -t undef

```
UUID                                          NODE              METHOD        ␣
↪MTM       SERIAL
51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB0  undef             undef         ␣
↪786310X   1052EF0
51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB1  undef             undef         ␣
↪786310X   1052EF1
```

3

Display all the discovered nodes:

nodediscoverls -t all

```
UUID                                          NODE              METHOD        ␣
↪MTM       SERIAL
51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB0  undef             undef         ␣
↪786310X   1052EF0
51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB1  undef             undef         ␣
↪786310X   1052EF1
51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB2  distest1          sequential    ␣
↪786310X   1052EF2
51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB3  distest2          sequential    ␣
↪786310X   1052EF3
```

4

Display the discovered node whose uuid is **51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB2**, with detailed information:

nodediscoverls -u 51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB2 -l

```
Object uuid: 51E5F2D7-0D59-11E2-A7BC-3440B5BEDBB2
   node=distest1
   method=sequential
   discoverytime=03-31-2013 17:05:12
   arch=x86_64
   cpucount=32
   cputype=Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz
   memory=198460852
   mtm=786310X
   serial=1052EF2
   nicdriver=eth0!be2net,eth1!be2net
   nicipv4=eth0!10.0.0.212/8
   nichwaddr=eth0!34:40:B5:BE:DB:B0,eth1!34:40:B5:BE:DB:B4
   nicpci=eth0!0000:0c:00.0,eth1!0000:0c:00.1
   nicloc=eth0!Onboard Ethernet 1,eth1!Onboard Ethernet 2
```

```
   niconboard=eth0!1,eth1!2
   nicfirm=eth0!ServerEngines BE3 Controller,eth1!ServerEngines BE3␣
→Controller
   switchname=eth0!c909f06sw01
   switchaddr=eth0!192.168.70.120
   switchdesc=eth0!IBM Flex System Fabric EN4093 10Gb Scalable Switch, flash␣
→image: version 7.2.6, boot image: version 7.2.6
   switchport=eth0!INTA2
```

SEE ALSO

nodediscoverstart.1

NAME

nodediscoverstart - starts the node discovery process

SYNOPSIS

nodediscoverstart [**-h** | **--help** | **-v** | **--version**]

Sequential Discovery Specific:

nodediscoverstart noderange=*noderange* [**hostiprange=***imageprofile*] [**bmciprange=***bmciprange*] [**groups=***groups*] [**rack=***rack*] [**chassis=***chassis*] [**height=***height*] [**unit=***unit*] [osimage=<osimagename>] [-n|-****-dns**] [**-s|--skipbmcsetup**] [****-V|--verbose**]

Profile Discovery Specific:

nodediscoverstart networkprofile=*network-profile* **imageprofile=***image-profile* **hostnameformat=***nost-name-format* [**hardwareprofile=***hardware-profile*] [**groups=***node-groups*] [**rack=***rack-name*] [**chassis=***chassis-name*] [**height=***rack-server-height*] [**unit=***rack-server-unit-location*] [**rank=***rank-num*]

DESCRIPTION

The **nodediscoverstart** command starts either the **Sequential Discovery** or **Profile Discovery** process. They can not both be running at the same time.

Sequential Discovery Specific:

This is the simplest discovery approach. You only need to specify the **noderange**, **hostiprange** and **bmciprange** that should be given to nodes that are discovered. (If you pre-define the nodes (via nodeadd or mkdef) and specify their host and BMC IP addresses, then you only need to specify the **noderange** to the **nodediscoverstart** command.) Once you have run **nodediscoverstart**, then physically power on the nodes in the sequence that you want them to receive the node names and IPs, waiting a short time (e.g. 30 seconds) between each node.

Profile Discovery Specific:

This is the PCM discovery approach. *networkprofile*, *imageprofile*, *hostnameformat* arguments must be specified to start the **Profile Discovery**. All nodes discovered by this process will be associated with specified profiles and rack/chassis/unit locations.

When the nodes are discovered, PCM updates the affected configuration files on the management node automatically. Configuration files include the /etc/hosts service file, DNS configuration, and DHCP configuration. Kit plug-ins are automatically triggered to update kit related configurations and services.

When you power on the nodes, they PXE boot and DHCP/TFTP/HTTP on the management node give each node the xCAT genesis boot image, which inventories the node hardware and sends data to the management node. There, either the sequential discovery process or the profile discovery process assigns node attributes and defines the node in the the database.

OPTIONS

noderange=noderange

> The set of node names that should be given to nodes that are discovered via the **Sequential Discovery** method. This argument is required to **Sequential Discovery**. Any valid xCAT **noderange** is allowed, e.g. node[01-10].

hostiprange=ip range

> The ip range which will be assigned to the host of new discovered nodes in the **Sequential Discovery** method. The format can be: *start_ip-end_ip* or *noderange*, e.g. 192.168.0.1-192.168.0.10 or 192.168.0.[1-10].

bmciprange=ip range

> The ip range which will be assigned to the bmc of new discovered nodes in the **Sequential Discovery** method. The format can be: *start_ip-end_ip* or *noderange*, e.g. 192.168.1.1-192.168.1.10 or 192.168.1.[1-10].

imageprofile=image-profile

> Sets the new image profile name used by the discovered nodes in the **Profile Discovery** method. An image profile defines the provisioning method, OS information, kit information, and provisioning parameters for a node. If the "__ImageProfile_imgprofile" group already exists in the nodehm table, then "imgprofile" is used as the image profile name.

networkprofile=network-profile

> Sets the new network profile name used by the discovered nodes in the **Profile Discovery** method. A network profile defines the network, NIC, and routes for a node. If the "__NetworkProfile_netprofile" group already exists in the nodehm table, then "netprofile" is used as the network profile name.

hardwareprofile=hardware-profile

> Sets the new hardware profile name used by the discovered nodes in the **Profile Discovery** method. If a "__HardwareProfile_hwprofile" group exists, then "hwprofile" is the hardware profile name. A hardware profile defines hardware management related information for imported nodes, including: IPMI, HMC, CEC, CMM.

hostnameformat=nost-name-format

> Sets the node name format for all discovered nodes in the **Profile Discovery** method. The two types of formats supported are prefix#NNNappendix and prefix#RRand#NNappendix, where wildcard #NNN and #NN are replaced by a system generated number that is based on the provisioning order. Wildcard #RR represents the rack number and stays constant.

For example, if the node name format is compute-#NN, the node name is generated as: compute-00, compute-01, ..., compute-99. If the node name format is blade#NNN-x64, the node name is generated as: blade001-x64, blade002-x64, ..., blade999-x64

For example, if the node name format is compute-#RR-#NN and the rack number is 2, the node name is generated as: compute-02-00, compute-02-01, ..., compute-02-99. If node name format is node-#NN-in-#RR and rack number is 1, the node name is generated as: node-00-in-01, node-01-in-01, ..., node-99-in-01

groups=*node-groups*

Sets the node groups that the discovered nodes should be put in for either the Sequential Discovery or Profile Discovery methods, where *node-group* is a comma-separated list of node groups.

rack=*rack-name>*

Sets the rack name where the node is located for either the Sequential Discovery or Profile Discovery methods.

chasiss=*chassis-name*

Sets the chassis name that the Blade server or PureFlex blade is located in, for either the Sequential Discovery or Profile Discovery methods. This option is used for the Blade server and PureFlex system only. You cannot specify this option with the rack option.

height=*rack-server-height*

Sets the height of a rack-mounted server in U units for either the Sequential Discovery or Profile Discovery methods. If the rack option is not specified, the default value is 1.

unit=*rack-server-unit-location*

Sets the start unit value for the node in the rack, for either the Sequential Discovery or Profile Discovery methods. This option is for a rack server only. If the unit option is not specified, the default value is 1

rank=*rank-num*

Specifies the starting rank number that is used in the node name format, for the Profile Discovery method. The rank number must be a valid integer between 0 and 254. This option must be specified with node-nameformat option. For example, if your node name format is compute-#RR-#NN. The rack's number is 2 and rank is specified as 5, the node name is generated as follows: compute-02-05, compute-02-06, ..., compute-02-99.

osimage=*osimagename*

Specifies the osimage name that will be associated with the new discovered node, the os provisioning will be started automatically at the end of the discovery process.

-n|--dns

Specifies to run makedns <nodename> for any new discovered node. This is useful mainly for non-predefined configuration, before running the "nodediscoverstart -n", the user needs to run makedns -n to initialize the named setup on the management node.

-s|--skipbmcsetup

Specifies to skip the bmcsetup during the sequential discovery process, if the bmciprange is specified with nodediscoverstart command, the BMC will be setup automatically during the discovery process, if the user does not want to run bmcsetup, could specify the "-s|--skipbmcsetup" with nodediscoverstart command to skip the bmcsetup.

-V|--verbose

Enumerates the free node names and host/bmc ips that are being specified in the ranges given. Use this option with Sequential Discovery to ensure that you are specifying the ranges you intend.

-h|--help

Display usage message.

-v|--version

Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occured.

EXAMPLES

1

Sequential Discovery: To discover nodes with noderange and host/bmc ip range:

nodediscoverstart noderange=n[1-10] hostiprange='172.20.101.1-172.20.101.10' bmciprange='172.20.102.1-172.20.102.10' -V

```
Sequential Discovery: Started:
   Number of free node names: 10
   Number of free host ips: 10
   Number of free bmc ips: 10
---------------------------------Free Nodes-------------------------------
↪-----
NODE                HOST IP             BMC IP
n01                 172.20.101.1        172.20.102.1
n02                 172.20.101.2        172.20.102.2
...                 ...                 ...
```

2

Profile Discovery: To discover nodes using the default_cn network profile and the rhels6.3_packaged image profile, use the following command:

nodediscoverstart networkprofile=default_cn imageprofile=rhels6.3_packaged hostnameformat=compute#NNN

SEE ALSO

nodediscoverstop(1)|nodediscoverstop.1, nodediscoverls(1)|nodediscoverls.1, nodediscoverstatus(1)|nodediscoverstatus.1

nodediscoverstatus.1

NAME

nodediscoverstatus - gets the node discovery process status

SYNOPSIS

nodediscoverstatus [**-h** | **--help** | **-v** | **--version**]

DESCRIPTION

The **nodediscoverstatus** command detects if the sequential or profile node discovery process is currently running, i.e. **nodediscoverstart**has been run, but **nodediscoverstop** has not.

OPTIONS

-h|--help

Display usage message.

-v|--version

Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occured.

EXAMPLES

To determine if there are some nodes discovered and the discovered nodes' status, enter the following command:

nodediscoverstatus

SEE ALSO

nodediscoverstart(1)|nodediscoverstart.1,   nodediscoverls(1)|nodediscoverls.1,   nodediscoversta-
tus(1)|nodediscoverstop.1

nodediscoverstop.1

NAME

nodediscoverstop - stops the node discovery process.

SYNOPSIS

nodediscoverstop [**-h** | **--help** | **-v** | **--version**]

DESCRIPTION

The **nodediscoverstop** command stops the sequential or profile node discovery process. Once this command has been run, newly discovered nodes will not be assigned node names and attributes automatically via the sequential or profile discovery process.

OPTIONS

-h|--help

Display usage message.

-v|--version

Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occured.

EXAMPLES

nodediscoverstop

SEE ALSO

nodediscoverstart(1)|nodediscoverstart.1,   nodediscoverls(1)|nodediscoverls.1,   nodediscoversta-tus(1)|nodediscoverstatus.1

nodegrpch.1

NAME

nodegrpch - Changes attributes at the group level in the xCAT cluster database.

SYNOPSIS

nodegrpch *group1,group2,... table.column=value* [...]

nodegrpch {**-v** | **--version**}

nodegrpch [**-?** | **-h** | **--help**]

DESCRIPTION

The nodegrpch command is similar to the nodech command, but ensures that the parameters are declared at the group level rather than the node specific level, and clears conflicting node specific overrides of the specified groups. Using table.column=value will do a verbatim assignment. If ",=" is used instead of "=", the specified value will be prepended to the attribute's comma separated list, if it is not already there. If "^=" is used, the specified value will be removed from the attribute's comma separated list, if it is there. You can also use "^=" and ",=" in the same command to essentially replace one item in the list with another. (See the Examples section.)

With these operators in mind, the unambiguous assignment operator is '=@'. If you need, for example, to set the nodehm.comments to =foo, you would have to do *nodegrpch group1 nodehm.comments=@=foo*.

See the **xcatdb** man page for an overview of each table.

The nodegrpch command also supports some short cut names as aliases to common attributes. See the **nodels** man page for details.

OPTIONS

-v|--version

> Command Version.

-?|-h|--help

> Display usage message.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

*

> To declare all members of ipmi group to have nodehm.mgt be ipmi

> ** nodegrpch** *ipmi nodehm.mgt=ipmi*

FILES

/opt/xcat/bin/nodegrpch

SEE ALSO

nodech(1)|nodech.1, nodels(1)|nodels.1, nodeadd(8)|nodeadd.8, noderange(3)|noderange.3

nodeimport.1

NAME

nodeimport - Create profiled nodes by importing hostinfo file.

SYNOPSIS

nodeimport [-h| -****-help | -v | -****-version]

nodeimport file=<hostinfo-filename> networkprofile=<network-profile> imageprofile=<image-profile> hostnameformat=<node-name-format> [hardwareprofile=<hardware-profile>] [groups=<node-groups>]

DESCRIPTION

The **nodeimport** command creates nodes by importing a hostinfo file which is following stanza format. In this hostinfo file, we can define node's hostname, ip, mac, switch name, switch port and host location infomation like rack, chassis, start unit, server height...etc

After nodes imported, the configuration files related with these nodes will be updated automatically. For example: /etc/hosts, dns configuration, dhcp configuration. And the kits node plugins will also be triggered automatically to update kit related configuration/services.

OPTIONS

-h|--help

Display usage message.

-v|--version

Command Version.

file=<nodeinfo-filename>

Specifies the node information file, where <nodeinfo-filename> is the full path and file name of the node information file.

imageprofile=<image-profile>

Sets the new image profile name used by the node, where <image-profile> is the new image profile. An image profile defines the provisioning method, OS information, kit information, and provisioning parameters for a node. If the "__ImageProfile_imgprofile" group already exists in the nodehm table, then "imgprofile" is used as the image profile name.

networkprofile=<network-profile>

Sets the new network profile name used by the node, where <network-profile> is the new network profile. A network profile defines the network, NIC, and routes for a node. If the "__NetworkProfile_netprofile" group already exists in the nodehm table, then "netprofile" is used as the network profile name.

hardwareprofile=<hardware-profile>

Sets the new hardware profile name used by the node, where <hardware-profile> is the new hardware management profile used by the node. If a "__HardwareProfile_hwprofile" group exists, then "hwprofile" is the hardware profile name. A hardware profile defines hardware management related information for imported nodes, including: IPMI, HMC, CEC, CMM.

hostnameformat=<host-name-format>

Sets the node name format for all nodes discovered, where <node-name-format> is a supported format. The two types of formats supported are prefix#NNNappendix and prefix#RRand#NNappendix, where wildcard #NNN and #NN are replaced by a system generated number that is based on the provisioning order. Wildcard #RR represents the rack number and stays constant.

For example, if the node name format is compute-#NN, the node name is generated as: compute-00, compute-01, ... , compute-99. If the node name format is blade#NNN-x64, the node name is generated as: blade001-x64, blade002-x64, ... , blade999-x64

For example, if the node name format is compute-#RR-#NN and the rack number is 2, the node name is generated as: compute-02-00, compute-02-01, ..., compute-02-99. If node name format is node-#NN-in-#RR and rack number is 1, the node name is generated as: node-00-in-01, node-01-in-01, ... , node-99-in-01

groups=<node-groups>

Sets the node groups that the imported node belongs to, where <node-group> is a comma-separated list of node groups.

RETURN VALUE

0 The command completed successfully.

1 An error has occured while validating parameters.

2 An error has occured while parsing hostinfo file.

EXAMPLES

To import nodes using a profile, follow the following steps:

1. Find all node groups and profiles, run the following command "tabdump nodegroups". For detailed profile information run "lsdef -t group <groupname>". Example of detailed profile information:

```
# tabdump nodegroup
#groupname,grouptype,members,membergroups,wherevals,comments,disable
"compute","static",,,,,
"__HardwareProfile_default_ipmi","static","static",,,,
"__NetworkProfile_default_mn","static","static",,,,
"__NetworkProfile_default_cn","static",,,,,
"__ImageProfile_rhels6.2-x86_64-install-compute","static","static",,,,

# lsdef -t group __NetworkProfile_default_cn
Object name: __NetworkProfile_default_cn
    grouptype=static
    installnic=eth0
    members=compute-000,compute-001
    netboot=xnba
    nichostnamesuffixes=eth0:-eth0
    nicnetworks=eth0:provision
    nictypes=eth0:Ethernet
    primarynic=eth0
```

2. Prepare a node information file.

```
Example of a node information file, a blade and a rack server defined:
# hostinfo begin
# This entry defines a blade.
```

```
__hostname__:
  mac=b8:ac:6f:37:59:24
  ip=192.168.1.20
  chassis=chassis01

# This entry defines a rack server.
__hostname__:
  mac=b8:ac:6f:37:59:25
  ip=192.168.1.20
  rack=rack01
  height=1
  unit=2

# hostinfo end.

Another example of a node infomation file, a PureFlex X/P node defined:
# hostinfo begin
# To define a PureFlex P/X node, chassis and slot id must be specified.
# The chassis must be a PureFlex chassis.
__hostname__:
  mac=b8:ac:6f:37:59:25
  chassis=cmm01
  slotid=1
# hostinfo end.

Example of a node information file, a switch auto discovery node defined:
# hostinfo begin
# This entry defines a blade.
__hostname__:
  switches=eth0!switch1!1,eth0!switch2!1!eth1

Example of a node information file that specifies a CEC-based rack-mounted Power node␣
↪that uses direct FSP management:
# Node information file begins
# This entry defines a Power rack-mount node.
__hostname__:
  mac=b8:ac:6f:37:59:28
  cec=mycec

__hostname__:
  mac=b8:ac:6f:37:59:28
  cec=mycec
  lparid=2
# Node information file ends.

Example of a node information file that specifies a PowerKVM Guest node that uses KVM␣
↪management:

# Node information file begins
# This entry defines a PowerKVM Guest node.
# Make sure the node 'vm01' is already created on Hypervisor
vm01:
  mac=b8:ef:3f:28:31:15
  vmhost=pkvm1
# Node information file ends.
```

The node information file includes the following items:

__hostname__: This is a mandatory item.

Description: The name of the node, where __hostname__ is automatically generated by the node name format. You can also input a fixed node name, for example "compute-node".

mac=<mac-address> This is a mandatory item.

Description: Specify the MAC address for the NIC used by the provisionging node, where <mac-address> is the NICs MAC address.

switches=<nic-name!switch-name!switch-port> This is a mandatory item, when define switch, switchport and node nic name relationship.

Description: Specify nic name, switch name and switch port to define node and switch relationship. We can define multi nic-switch-port relations here, looks like: switches=eth0!switch1!1,eth1!switch1,2

slotid=<slot-id> This is a mandatory item while define a PureFlex node.

Description: The node position in the PureFlex Chassis.

cec=<cec-name> This is a mandatory option for defining Power rack-mounted nodes.

Description: Specifies the name of a Power rack-mount central electronic complex (CEC).

lparid=<lpar-id> This is a optional option for defining Power rack-mounted nodes.

Description: Specifies the LPAR ID of a Power rack-mounted node, where <lpar-id> is the ID number. The default value is 1 if it is not defined.

ip=<ip-address> This is an optional item.

Description: Specify the IP address used for provisioning a node, where <ip-address> is in the form xxx.xxx.xxx.xxx. If this item is not included, the IP address used to provision the node is generated automatically according to the Network Profile used by the node.

nicips=<nics-ip> This is an optional item.

Description: Lists the IP address for each network interface configuration (NIC) used by the node, excluding the provisioning network, where <nics-ip> is in the form <nic1>!<nic-ip1>,<nic2>!<nic-ip2>,.... For example, if you have 2 network interfaces configured, the nicips attribute should list both network interfaces: nicips=eth1!10.10.10.11,bmc!192.168.10.3. If the nicips attribute is not specified, the IP addresses are generated automatically according to the network profile.

rack=<rack-name> This is an optional item.

Description: node location info. Specify the rack name which this node will be placed into. If not specify this item, there will be no node location info set for this node. this item must be specified together with height + unit.

chasiss=<chassis-name> This is an optional item.

Description: node location info, for blade(or PureFlex) only. Specify the chasiss name which this blade will be placed into. this item can not be specified together with rack.

height=<chassis-height> This is an optional item.

Description: node location info, for rack server only. Specify the server height number, in U. this item must be specified together with rack and unit.

unit=<rack-server-unit-location> This is an optional item.

Description: node location info, for rack server only. Specify the node's start unit number in rack, in U. this item must be specified together with rack and height.

vmhost=<PowerKVM Hypervisior Host Name> This is a mandatory option for defining PowerKVM Guest nodes.

Description: Specifies the vmhost of a Power KVM Guest node, where <vmhost> is the host name of PowerKVM Hypervisior.

3. Import the nodes, by using the following commands. Note: If we want to import PureFlex X/P nodes, hardware profile must be set to a PureFlex hardware type.

   nodeimport   file=/root/hostinfo.txt   networkprofile=default_cn   imageprofile=rhels6.3_packaged hostnameformat=compute-#NNN

4. After importing the nodes, the nodes are created and all configuration files used by the nodes are updated, including: /etc/hosts, DNS, DHCP.

5. Reboot the nodes. After the nodes are booted they are provisioned automatically.

SEE ALSO

nodepurge(1)|nodepurge.1, nodechprofile(1)|nodechprofile.1, noderefresh(1)|noderefresh.1

nodels.1

NAME

nodels - lists the nodes, and their attributes, from the xCAT database.

SYNOPSIS

nodels [*noderange*] [**-b** | **--blame**] [**-H** | **--with-fieldname**] [**-S**] [*table.column* | *shortname*] [...]

nodels [*noderange*] [**-H** | **--with-fieldname**] [*table*]

nodels [**-?** | **-h** | **--help** | **-v** | **--version**]

DESCRIPTION

The **nodels** command lists the nodes specified in the node range. If no noderange is provided, then all nodes are listed.

Additional attributes of the nodes will also be displayed if the table names and attribute names are specified after the noderange in the form: *table.column* . A few shortcut names can also be used as aliases to common attributes:

groups

> nodelist.groups

tags

> nodelist.groups

mgt

> nodehm.mgt

nodels can also select based on table value criteria. The following operators are available:

==

> Select nodes where the table.column value is exactly a certain value.

!=

Select nodes where the table.column value is not a given specific value.

=~

Select nodes where the table.column value matches a given regular expression.

!~

Select nodes where the table.column value does not match a given regular expression.

The **nodels** command with a specific node and one or more table.attribute parameters is a good substitute for grep'ing through the tab files, as was typically done in xCAT 1.x. This is because nodels will translate any regular expression rows in the tables into their meaning for the specified node. The tab* commands will not do this, instead they will just display the regular expression row verbatim.

OPTIONS

-v|--version

Command Version.

-H|--with-fieldname

Force display of table name and column name context for each result

-b|--blame

For values inherited from groups, display which groups provided the inheritence

-S

List all the hidden nodes (FSP/BPA nodes) with other ones.

-?|-h|--help

Display usage message.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To list all defined nodes, enter:

```
nodels
```

Output is similar to:

```
node1
node2
node3
```

2. To list all defined attributes in a table for a node or noderange, enter:

```
nodels rra001a noderes
```

Output is similar to:

```
rra001a: noderes.primarynic: eth0
rra001a: noderes.xcatmaster: rra000
rra001a: noderes.installnic: eth0
rra001a: noderes.netboot: pxe
rra001a: noderes.servicenode: rra000
rra001a: noderes.node: rra001a
```

3. To list nodes in node group ppc, enter:

```
nodels ppc
```

Output is similar to:

```
ppcnode1
ppcnode2
ppcnode3
```

4. To list the groups each node is part of:

```
nodels all groups
```

Output is similar to:

```
node1: groups: all
node2: groups: all,storage
node3: groups: all,blade
```

5. To list the groups each node is part of:

```
nodels all nodehm.power
```

Output is similar to:

```
node1: nodehm.power: blade
node2: nodehm.power: ipmi
node3: nodehm.power: ipmi
```

6. To list the out-of-band mgt method for blade1:

```
nodels blade1 nodehm.mgt
```

Output is similar to:

```
blade1: blade
```

7. Listing blades managed through an AMM named 'amm1'

```
nodels all mp.mpa==amm1
```

Output is similar to:

```
blade1
blade10
blade11
blade12
blade13
```

```
blade2
blade3
blade4
blade5
blade6
blade7
blade8
blade9
```

8. Listing the switch.switch value for nodes in the second rack:

```
nodels all nodepos.rack==2 switch.switch
```

Output is similar to:

```
n41: switch.switch: switch2
n42: switch.switch: switch2
n43: switch.switch: switch2
n44: switch.switch: switch2
n45: switch.switch: switch2
n46: switch.switch: switch2
n47: switch.switch: switch2
n55: switch.switch: switch2
n56: switch.switch: switch2
n57: switch.switch: switch2
n58: switch.switch: switch2
n59: switch.switch: switch2
n60: switch.switch: switch2
```

9. Listing the blade slot number for anything managed through a device with a name beginning with amm:

```
nodels all mp.mpa=~/^amm.*/ mp.id
```

Output looks like:

```
blade1: mp.id: 1
blade10: mp.id: 10
blade11: mp.id: 11
blade12: mp.id: 12
blade13: mp.id: 13
blade2: mp.id: 2
blade3: mp.id: 3
blade4: mp.id: 4
blade5: mp.id: 5
blade6: mp.id: 6
blade7: mp.id: 7
blade8: mp.id: 8
blade9: mp.id: 9
```

10. To list the hidden nodes that can't be seen with other flags. The hidden nodes are FSP/BPAs.

```
lsdef -S
```

FILES

/opt/xcat/bin/nodels

SEE ALSO

noderange(3)|noderange.3, tabdump(8)|tabdump.8, lsdef(1)|lsdef.1

nodepurge.1

NAME

nodepurge - Removes nodes.

SYNOPSIS

nodepurge [-h| -****-help | -v | -****-version]

nodepurge <noderange>

DESCRIPTION

The **nodepurge** automatically removes all nodes from the database and any related configurations used by the node.

After the nodes are removed, the configuration files related to these nodes are automatically updated, including the following files: /etc/hosts, DNS, DHCP. Any kits that are used by the nodes are triggered to automatically update kit configuration and services.

OPTIONS

-h|--help

Display usage message.

-v|--version

Command Version

noderange

The nodes to be removed.

RETURN VALUE

0 The command completed successfully.

1 An error has occured.

EXAMPLES

To remove nodes compute-000 and compute-001, use the following command:

nodepurge compute-000,compute-001

SEE ALSO

nodeimport(1)|nodeimport.1,   nodechprofile(1)|nodechprofile.1,   noderefresh(1)|noderefresh.1, noderange(3)|noderange.3

noderefresh.1

NAME

noderefresh - Update nodes configurations by running associated kit plugins.

SYNOPSIS

noderefresh [-h| -****-help | -v | -****-version]

noderefresh <noderange>

DESCRIPTION

The **noderefresh** command will update nodes settings, it will call all associated kit plug-in configurations and also services

OPTIONS

-h|--help

Display usage message.

-v|--version

Command Version.

noderange

The nodes to be updated.

RETURN VALUE

0 The command completed successfully.

1 An error has occured.

EXAMPLES

noderefresh compute-000,compute-001

SEE ALSO

nodeimport(1)|nodeimport.1, nodechprofile(1)|nodechprofile.1, nodepurge(1)|nodepurge.1, noderange(3)|noderange.3

---

noderm.1

NAME

noderm -Removes the nodes in the noderange from all database table.

SYNOPSIS

noderm [-h| –help]

noderm noderange

DESCRIPTION

```
The noderm command removes the nodes in the input node range.
```

OPTIONS

-h Display usage message.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

    1. To remove the nodes in noderange node1-node4, enter:

noderm node1-node4

FILES

/opt/xcat/bin/noderm

SEE ALSO

nodels(1)|nodels.1, nodeadd(8)|nodeadd.8, noderange(3)|noderange.3

nodestat.1

Name

nodestat - display the running status of each node in a noderange

Synopsis

nodestat [*noderange*] [-*m*|-*-**usemon***] [*-**p***|*--**powerstat***] [*-**f***] [*-**u***|*-******-updatedb*]

nodestat [-*h*|-*-**help***|*-**v***|*-******-version*]

Description

nodestat displays and optionally updates the database the running status of a single or range of nodes or groups. See noderange(3)|noderange.3.

By default, it works as following:

1. gets the sshd,pbs_mom,xend port status;

2. if none of them are open, it gets the fping status;

3. for pingable nodes that are in the middle of deployment, it gets the deployment status;

4. for non-pingable nodes, it shows 'noping'.

When -m is specified and there are settings in the monsetting table, it displays the status of the applications specified in the monsetting table. When -p is specified it shows the power status for the nodes that are not pingable. When -u is specified it saves the status info into the xCAT database. Node's pingable status and deployment status is saved in the nodelist.status column. Node's application status is saved in the nodelist.appstatus column.

To specify settings in the **monsetting** table, use 'xcatmon' as the name, 'apps' as the key and the value will be a list of comma separated list of application names. For each application, you can specify the port number that can be queried on the nodes to get the running status. Or you can specify a command that can be called to get the node status from. The command can be a command that can be run locally at the management node or the service node for hierarchical cluster, or a command that can be run remotely on the nodes.

The following is an example of the settings in the **monsetting** table:

```
name key value
xcatmon apps ssh,ll,gpfs,someapp
xcatmon gpfs cmd=/tmp/mycmd,group=compute,group=service
xcarmon ll port=9616,group=compute
xcatmon someapp dcmd=/tmp/somecmd
```

Keywords to use:

```
apps -- a list of comma separated application names whose status will be queried. For
→how to get the status of each app, look for app name in the key filed in a
→different row.
port -- the application daemon port number, if not specified, use internal list, then
→/etc/services.
group -- the name of a node group that needs to get the application status from. If
→not specified, assume all the nodes in the nodelist table. To specify more than one
→groups, use group=a,group=b format.
cmd -- the command that will be run locally on mn or sn.
lcmd -- the command that will be run the the mn only.
dcmd -- the command that will be run distributed on the nodes using xdsh <nodes> ....
```

For commands specified by 'cmd' and 'lcmd', the input of is a list of comma separated node names, the output must be in the following format:

```
node1:string1
node2:string2
...
```

For the command specified by 'dcmd', no input is needed, the output can be a string.

Options

-f

> Uses fping instead of nmap even if nmap is available. If you seem to be having a problem with false
> negatives, fping can be more forgiving, but slower.

-m | --usemon

> Uses the settings from the **monsetting** talbe to determine a list of applications that need to get status for.

-p | --powerstat

> Gets the power status for the nodes that are 'noping'.

-u | --updatedb

> Updates the status and appstatus columns of the nodelist table with the returned running status from the
> given nodes.

-v | --version

> Print version.

-h | --help

> Print help.

Examples

1. nodestat compute

```
node1    sshd
node2    sshd
node3    ping
node4    pbs
node5    noping
```

2. nodestat compute -p

```
node1    sshd
node2    sshd
node3    ping
node4    pbs
node5    noping(Shutting down)
```

3. nodestat compute -u

   node1 sshd node2 sshd node3 ping node4 netboot node5 noping

4. nodestat compute -m

   node1 ping,sshd,ll,gpfs=ok node2 ping,sshd,ll,gpfs=not ok,someapp=something is wrong node3 netboot
   node4 noping

See Also

noderange(3)|noderange.3, nodels(1)|nodels.1, nodeset(8)|nodeset.8

packimage.1

NAME

packimage - Packs the stateless image from the chroot file system.

SYNOPSIS

packimage [-h| –help]

packimage [-v| –version]

packimage [-o OS] [-p profile] [-a architecture] [-m method]

packimage imagename

DESCRIPTION

Packs the stateless image from the chroot file system into a file system to be sent to the node for a diskless install. The install dir is setup by using "installdir" attribute set in the site table. The nodetype table "profile" attribute for the node should reflect the profile of the install image.

If no *imagename* is specified, this command uses the os image definition files from /install/custom/netboot/[os] directory first. If not found, it falls back to the default directory /opt/xcat/share/xcat/netboot/[os]. If a *imagename* is specified, this command will get all the necessary os image definition files from the *osimage* and *linuximage* tables.

Parameters

imagename specifies the name of a os image definition to be used. The specification for the image is stored in the *osimage* table and *linuximage* table.

OPTIONS

-h Display usage message.

-v Command Version.

-o Operating system (fedora8, rhel5, sles10,etc)

-p Profile (compute,service)

-a Architecture (ppc64,x86_64,etc)

-m Method (default cpio)

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To pack a fedora8 image for a compute node architecture x86_64 and place it in the /install/netboot/fedora8/x86_64/compute/rootimg.gz file enter:

packimage -o fedora8 -p compute -a x86_64

This would use the package information from the /install/custom/netboot/fedora/compute* files first. If not found it uses /opt/xcat/share/xcat/netboot/fedora/compute* files.

FILES

/opt/xcat/sbin/packimage

NOTES

This command is part of the xCAT software product.

SEE ALSO

genimage(1)|genimage.1

pasu.1

NAME

pasu - run the ASU to many nodes in parallel

SYNOPSIS

pasu [**-V**] [**-d**] [**-n**] [**-l** *user*] [**-p** *passwd*] [**-f** *fanout*] [**-i** *hostname-suffix*] *noderange command*

pasu [**-V**] [**-d**] [**-n**] [**-l** *user*] [**-p** *passwd*] [**-f** *fanout*] [**-i** *hostname-suffix*] **-b** *batchfile noderange*

pasu [**-h** | **--help**]

DESCRIPTION

The **pasu** command runs the ASU command in out-of-band mode in parallel to multiple nodes. Out-of-band mode means that ASU connects from the xCAT management node to the IMM (BMC) of each node to set or query the ASU settings. To see all of the ASU settings available on the node, use the "show all" command. To query or set multiple values, use the **-b** (batch) option. To group similar output from multiple nodes, use xcoll(1)|xcoll.1.

Before running **pasu**, you must install the ASU RPM from IBM. You can download it from the IBM Fix Central site. You also must configure the IMMs properly according to xCAT documentation. Run "**rpower** *noderange* **stat**" to confirm that the IMMs are configured properly.

OPTIONS

-n|--nonodecheck

> Do not send the noderange to xcatd to expand it into a list of nodes. Use the noderange exactly as it is specified to pasu. In this case, the noderange must be a simple list of comma-separated hostnames of the IMMs.

-l|--loginname *username*

> The username to use to connect to the IMMs. If not specified, the row in the xCAT **passwd** table with key "ipmi" will be used to get the username.

-p|--passwd *passwd*

> The password to use to connect to the IMMs. If not specified, the row in the xCAT passwd table with key "ipmi" will be used to get the password.

-f|--fanout

> How many processes to run in parallel simultaneously. The default is 64. You can also set the XCATPSHFANOUT environment variable.

-b|--batch *-batchfile*

> A simple text file that contains multiple ASU commands, each on its own line.

-d|--donotfilter

> By default, pasu filters out (i.e. does not display) the standard initial output from ASU:

```
IBM Advanced Settings Utility version 9.30.79N
Licensed Materials - Property of IBM
(C) Copyright IBM Corp. 2007-2012 All Rights Reserved
Connected to IMM at IP address node2-imm
```

> If you want this output to be displayed, use this flag.

-i|--interface *hostname-suffix*

> The hostname suffix to be appended to the node names.

-V|--verbose

> Display verbose messages.

-h|--help

> Display usage message.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To display the Com1ActiveAfterBoot setting on 2 nodes:

```
pasu node1,node2 show DevicesandIOPorts.Com1ActiveAfterBoot
```

   Output is similar to:

```
node1: DevicesandIOPorts.Com1ActiveAfterBoot=Enable
node2: DevicesandIOPorts.Com1ActiveAfterBoot=Enable
```

2. To display the Com1ActiveAfterBoot setting on all compute nodes:

```
pasu compute show DevicesandIOPorts.Com1ActiveAfterBoot | xcoll
```

   Output is similar to:

```
====================================
compute
====================================
DevicesandIOPorts.Com1ActiveAfterBoot=Enable
```

3. To set several settings on all compute nodes, create a batch file called (for example) asu-settings with contents:

```
set DevicesandIOPorts.Com1ActiveAfterBoot Enable
set DevicesandIOPorts.SerialPortSharing Enable
set DevicesandIOPorts.SerialPortAccessMode Dedicated
set DevicesandIOPorts.RemoteConsole Enable
```

   Then run:

```
pasu -b asu-settings compute | xcoll
```

   Output is similar to:

```
====================================
compute
====================================
Batch mode start.
[set DevicesandIOPorts.Com1ActiveAfterBoot Enable]
DevicesandIOPorts.Com1ActiveAfterBoot=Enable

[set DevicesandIOPorts.SerialPortSharing Enable]
DevicesandIOPorts.SerialPortSharing=Enable

[set DevicesandIOPorts.SerialPortAccessMode Dedicated]
DevicesandIOPorts.SerialPortAccessMode=Dedicated

[set DevicesandIOPorts.RemoteConsole Enable]
DevicesandIOPorts.RemoteConsole=Enable

Beginning intermediate batch update.
Waiting for command completion status.
Command completed successfully.
Completed intermediate batch update.
Batch mode competed successfully.
```

4. To confirm that all the settings were made on all compute nodes, create a batch file called (for example) asu-show with contents:

```
show DevicesandIOPorts.Com1ActiveAfterBoot
show DevicesandIOPorts.SerialPortSharing
show DevicesandIOPorts.SerialPortAccessMode
show DevicesandIOPorts.RemoteConsole
```

Then run:

```
pasu -b asu-show compute | xcoll
```

Output is similar to:

```
====================================
compute
====================================
Batch mode start.
[show DevicesandIOPorts.Com1ActiveAfterBoot]
DevicesandIOPorts.Com1ActiveAfterBoot=Enable

[show DevicesandIOPorts.SerialPortSharing]
DevicesandIOPorts.SerialPortSharing=Enable

[show DevicesandIOPorts.SerialPortAccessMode]
DevicesandIOPorts.SerialPortAccessMode=Dedicated

[show DevicesandIOPorts.RemoteConsole]
DevicesandIOPorts.RemoteConsole=Enable

Batch mode competed successfully.
```

FILES

/opt/xcat/bin/pasu

SEE ALSO

noderange(3)|noderange.3, rpower(1)|rpower.1, xcoll(1)|xcoll.1

pcons.1

SYNOPSIS

pcons *noderange command*

pcons[**-h | --help**]

pcons[**-v | --version**]

DESCRIPTION

Runs the a command to the noderange using the console.

EXAMPLES

```
pcons 1,3 stat
pcons all,-129-256 stat
```

SEE ALSO

psh(1)|psh.1

pgsqlsetup.1

NAME

pgsqlsetup - Sets up the PostgreSQL database for xCAT to use.

SYNOPSIS

pgsqlsetup {**-h | --help**}

pgsqlsetup {**-v | --version**}

pgsqlsetup {**-i | --init**} [-N|nostart] [-P|-****-PCM]** [**-o|--setupODBC]** [****-V | --verbose**]

pgsqlsetup {**-o | --setupODBC**} [-V|-****-verbose]

DESCRIPTION

pgsqlsetup - Sets up the PostgreSQL database for xCAT to use. The pgsqlsetup script is run on the Management Node as root after the PostgreSQL code has been installed. The xcatd daemon will be stopped during migration. No xCAT commands should be run during the init process, because we will be migrating the xCAT database to PostgreSQL and restarting the xcatd daemon as well as the PostgreSQL daemon. For full information on all the steps that will be done reference One password must be supplied for the setup, a password for the xcatadm unix id and the same password for the xcatadm database id. The password will be prompted for interactively or you can set the XCATPGPW environment variable to the password and then there will be no prompt.

OPTIONS

-h|--help

    Displays the usage message.

-v|--version

    Displays the release version of the code.

-V|--verbose

    Displays verbose messages.

-i|--init

The init option is used to setup an installed PostgreSQL database so that xCAT can use the database. This involves creating the xcat database, the xcat admin id, allowing access to the xcatdb database by the Management Node. It customizes the postgresql.conf configuration file, adds the management server to the pg_hba.conf and starts the PostgreSQL server. It also backs up the current xCAT database and restores it into the newly setup xcatdb PostgreSQL database. It creates the /etc/xcat/cfgloc file to point the xcatd daemon to the PostgreSQL database and restarts the xcatd daemon using the database. On AIX, it additionally setup the xcatadm unix id and the postgres id and group. For AIX, you should be using the PostgreSQL rpms available from the xCAT website. For Linux, you should use the PostgreSQL rpms shipped with the OS. You can chose the -o option, to run after the init. To add additional nodes to access the PostgreSQL server, setup on the Management Node, edit the pg_hba.conf file.

For more documentation see:Setting_Up_PostgreSQL_as_the_xCAT_DB

-N|--nostart

This option with the -i flag will create the database, but will not backup and restore xCAT tables into the database. It will create the cfgloc file such that the next start of xcatd will try and contact the database. This can be used to setup the xCAT PostgreSQL database during or before install.

-P|--PCM

This option sets up PostgreSQL database to be used with xCAT running with PCM.

-o|--odbc

This option sets up the ODBC /etc/../odbcinst.ini, /etc/../odbc.ini and the .odbc.ini file in roots home directory will be created and initialized to run off the xcatdb PostgreSQL database.

ENVIRONMENT VARIABLES

XCATPGPW

The password to be used to setup the xCAT admin id for the database.

EXAMPLES

*

To setup PostgreSQL for xCAT to run on the PostgreSQL xcatdb database :

pgsqlsetup *-i*

*

To setup the ODBC for PostgreSQL xcatdb database access :

pgsqlsetup *-o*

pping.1

SYNOPSIS

pping [**-i** | **--interface** *interfaces*] [**-f** | **--use_fping**] *noderange*

pping [**-h** | **--help**]

pping {**-v** | **--version**}

DESCRIPTION

pping is a utility used to ping a list of nodes in parallel. **pping** will return an unsorted list of nodes with a ping or noping status. **pping** front-ends nmap or fping if available.

This command does not support the xcatd client/server communication. It must be run on the management node.

OPTIONS

-i | --interface *interfaces*

> A comma separated list of network interface names that should be pinged instead of the interface repre-sented by the nodename/hostname. The following name resolution convention is assumed: an interface is reachable by the hostname <nodename>-<interface>. For example, the ib2 interface on node3 has a hostname of node3-ib2.

> If more than one interface is specified, each interface will be combined with the nodenames as described above and will be pinged in turn.

-f | --use_fping

> Use fping instead of nmap

-h | --help

> Show usage information.

-v | --version

> Display the installed version of xCAT.

EXAMPLES

1. pping all

```
node1: ping
node2: ping
node3: noping
```

2. pping all -i ib0,ib1

```
node1-ib0: ping
node2-ib0: ping
node3-ib0: noping
node1-ib1: ping
node2-ib1: ping
node3-ib1: noping
```

SEE ALSO

psh(1)|psh.1, noderange(3)|noderange.3

ppping.1

SYNOPSIS

ppping [**-i** | **--interface** *interfaces*] [**-d** | **--debug**] [**-V** | **--verbose**] [**-q** | **--quiet**] [**-s** | **--serial**] *noderange*

ppping [**-h** | **--help**]

pping {**-v** | **--version**}

DESCRIPTION

ppping is a utility used to test the connectivity between nodes in the noderange using ping. By default, **ppping** will return an unsorted list of the node pairs that are not able to ping each other, or a message that all nodes are pingable. More or less output can be controlled by the -V and -q options. **ppping** front-ends **pping** and **xdsh**.

OPTIONS

-s

> Ping serially instead of in parallel.

-i | **--interface** *interfaces*

> A comma separated list of network interface names that should be pinged instead of the interface represented by the nodename/hostname. The following name resolution convention is assumed: an interface is reachable by the hostname <nodename>-<interface>. For example, the ib2 interface on node3 has a hostname of node3-ib2.

> If more than one interface is specified, each interface will be combined with the nodenames as described above and will be pinged in turn.

-V | **--verbose**

> Display verbose output. The result of every ping attempt from every node will be displayed. Without this option, just a summary of the successful pings are displayed, along with all of the unsuccessful pings.

-q | **--quiet**

> Display minimum output: just the unsuccessful pings. This option has the effect that if all pings are successful, nothing is displayed. But it also has the performance benefit that each node does not have to send successful ping info back to the management node.

-d | **--debug**

> Print debug information.

-h | **--help**

> Show usage information.

-v | **--version**

> Display the installed version of xCAT.

EXAMPLES

1. ppping all -q

```
blade7: node2: noping
blade8: node2: noping
blade9: node2: noping
devmaster: node2: noping
node2: noping
```

2. ppping node1,node2 -i ib0,ib1,ib2,ib3

```
node1: pinged all nodes successfully on interface ib0
node1: pinged all nodes successfully on interface ib1
node1: pinged all nodes successfully on interface ib2
node1: pinged all nodes successfully on interface ib3
node2: pinged all nodes successfully on interface ib0
node2: pinged all nodes successfully on interface ib1
node2: pinged all nodes successfully on interface ib2
node2: pinged all nodes successfully on interface ib3
```

SEE ALSO

psh(1)|psh.1, pping(1)|pping.1

prsync.1

Name

prsync - parallel rsync

Synopsis

prsync *filename* [*filename ...*] *noderange:destinationdirectory*

prsync [*-o rsync options*] [**-f** *fanout*] [*filename filename ...*] [*directory directory ...*] *noderange:destinationdirectory*

prsync {**-h** | **--help** | **-v** | **--version**}

Description

prsync is a front-end to rsync for a single or range of nodes and/or groups in parallel.

Note: this command does not support the xcatd client/server communication and therefore must be run on the management node. It does not support hierarchy, use xdcp -F to run rsync from the management node to the compute node via a service node

prsync is NOT multicast, but is parallel unicasts.

Options

rsyncopts

> rsync options. See **rsync(1)**.

-f *fanout*

> Specifies a fanout value for the maximum number of concur- rently executing remote shell processes.

filename

> A space delimited list of files to rsync.

directory

> A space delimited list of directories to rsync.

noderange:destination

> A noderange(3)|noderange.3 and destination directory. The : is required.

-h | --help

> Print help.

-v | --version

> Print version.

XCATPSHFANOUT

> Specifies the fanout value. This variable is overridden by the **-f** flag. Default is 64.

Examples

*

> **cd** */install;* **prsync -o "crz"** *post stage:/install*

*

> **prsync** *passwd group rack01:/etc*

See Also

noderange(3)|noderange.3, pscp(1)|pscp.1, pping(1)|pping.1, psh(1)|psh.1

pscp.1

Name

pscp - parallel remote copy

Synopsis

pscp [-i *suffix*] [*scp options* ...] [**-f** *fanout*] *filename* [*filename* ...] *noderange:destinationdirectory*

pscp {**-h | --help | -v | --version**}

Description

pscp is a utility used to copy a single or multiple set of files and/or directories to a single or range of nodes and/or groups in parallel.

pscp is a front-end to the remote copy **scp**.

Note: this command does not support the xcatd client/server communication and therefore must be run on the management node. It does not support hierarchy, use xdcp to run remote copy command from the management node to the compute node via a service node.

pscp is NOT multicast, but is parallel unicasts.

Options

-f *fanout*

Specifies a fanout value for the maximum number of concur- rently executing remote shell processes.

-i *suffix*

Interfaces to be used.

scp options

See **scp(1)**

filename

A space delimited list of files to copy. If **-r** is passed as an scp option, directories may be specified as well.

noderange:destination

A noderange(3)|noderange.3 and destination directory. The : is required.

-h | --help

Print help.

-v | --version

Print version.

XCATPSHFANOUT

Specifies the fanout value. This variable is overridden by the **-f** flag. Default is 64.

Examples

pscp -r */usr/local node1,node3:/usr/local***pscp** *passwd group rack01:/etc*

See Also

noderange(3)|noderange.3, pping(1)|pping.1, prsync(1)|prsync.1, psh(1)|psh.1

psh.1

Name

psh - parallel remote shell

Synopsis

psh [**-i** *interface*] [**-f** *fanout*] [**-l** *user*] *noderange command*

psh {**-h** | **--help** | **-v** | **--version**}

Description

psh is a utility used to run a command across a list of nodes in parallel.

ssh must be set up to allow no prompting for **psh** to work.

Note:

This command does not run through xcatd like most xCAT commands do. This means you must either run it on the management node, or have a network connection between your machine and the nodes. It does not support hierarchy, use xdsh to run remote command from the management node to the compute node via a service node.

psh arguments need to precede noderange, otherwise, you will get unexpected errors.

Options

-i *interface*

> The NIC on the node that psh should communicate with. For example, if *interface* is **eth1**, then psh will concatenate **-eth1** to the end of every node name before ssh'ing to it. This assumes those host names have been set up to resolve to the IP address of each of the eth1 NICs.

-f *fanout*

> Specifies a fanout value for the maximum number of concur- rently executing remote shell processes.

-l *user*

> Log into the nodes as the specified username. The default is to use the same username as you are running the psh command as.

-n|--nonodecheck

> Do not send the noderange to xcatd to expand it into a list of nodes. Instead, use the noderange exactly as it is specified. In this case, the noderange must be a simple list of comma-separated hostnames of the nodes. This allows you to run **psh** even when xcatd is not running.

noderange

> See noderange(3)|noderange.3.

command

> Command to be run in parallel. If no command is give then **psh**enters interactive mode. In interactive mode a ">" prompt is displayed. Any command entered is executed in parallel to the nodes in the noderange. Use "exit" or "Ctrl-D" to end the interactive session.

-h | --help

> Print help.

Environment Variables

XCATPSHFANOUT

> Specifies the fanout value. This variable is overridden by the **-f** flag. Default is 64.

Examples

*

> Run uptime on 3 nodes:
>
> **psh** *node4-node6 uptime*
>
> node4: Sun Aug 5 17:42:06 MDT 2001 node5: Sun Aug 5 17:42:06 MDT 2001 node6: Sun Aug 5 17:42:06 MDT 2001

*

> Run a command on some BladeCenter management modules:
>
> **psh** *amm1-amm5 'info -T mm[1]'*

*

> Remove the tmp files on the nodes in the 1st frame:
>
> **psh** *rack01 'rm -f /tmp/*'*
>
> Notice the use of '' to forward shell expansion. This is not necessary in interactive mode.

See Also

noderange(3)|noderange.3, pscp(1)|pscp.1, pping(1)|pping.1, prsync(1)|prsync.1

pushinitrd.1

NAME

pushinitrd - queries your SoftLayer account and gets attributes for each server.

SYNOPSIS

pushinitrd [**-v | --verbose**] [**-w** *waittime*] [*noderange*]

pushinitrd [**-? | -h | --help**]

DESCRIPTION

The **pushinitrd** command copies the initrd, kernel, params, and static IP info to nodes, so they can be net installed even across vlans (w/o setting up pxe/dhcp broadcast relay). This assumes a working OS is on the nodes. Before running this command, you must run nodeset for these nodes. All of the nodes given to one invocation of **pushinitrd** must be using the same osimage.

Before using this command, if will be most convenient if you exchange the ssh keys using:

```
xdsh <noderange> -K
```

OPTIONS

-w *waittime*

> The number of seconds the initrd should wait before trying to communicate over the network. The default is 75. This translates into the netwait kernel parameter and is usually needed in a SoftLayer environment because it can take a while for a NIC to be active after changing state.

-?|-h|--help

> Display usage message.

-v|--version

> Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. Configure nodes for net installing in a SoftLayer environment:

   ```
   pushinitrd <noderange>
   ```

FILES

/opt/xcat/bin/pushinitrd

SEE ALSO

getslnodes(1)|getslnodes.1

rbeacon.1

SYNOPSIS

rbeacon *noderange* {**on | blink | off | stat**}

rbeacon [**-h | --help**]

rbeacon {**-v | --version**}

DESCRIPTION

rbeacon Turns beacon (a light on the front of the physical server) on/off/blink or gives status of a node or noderange.

EXAMPLES

```
rbeacon 1,3 off
rbeacon 14-56,70-203 on
rbeacon 1,3,14-56,70-203 blink
rbeacon all,-129-256 stat
```

SEE ALSO

noderange(3)|noderange.3, rpower(1)|rpower.1

rbootseq.1

SYNOPSIS

rbootseq [**-h | --help | -v | --version**]

Blade specific:

rbootseq *noderange* {**hd0 | hd1 | hd2 | hd3 | net | iscsi | iscsicrit | cdrom | usbflash | floppy | none | list | stat**},...

HP Blade specific:

rbootseq *noderange* {**hd | net1 | net2 | net3 | net4 | cdrom | usbflash | floppy | none**},...

PPC (using Direct FSP Management) specific:

rbootseq *noderange* [**hfi|net**]

DESCRIPTION

For Blade specific:

rbootseq sets the boot sequence (the order in which boot devices should be tried) for the specified blades. Up to four different medium/devices can be listed, separated by commas. The boot sequence will remain in effect for these blades until set differently.

For PPC (using Direct FSP Management) specific:

rbootseq sets the ethernet (net) or hfi device as the first boot device for the specified PPC LPARs. The **rbootseq** command requires that the ethernet or hfi mac address is stored in the mac table, and that the network information is correct in the networks table.

OPTIONS

hd0 | harddisk0 | hd | harddisk

> The first hard disk.

hd1 | harddisk1

> The second hard disk.

hd2 | harddisk2

> The third hard disk.

hd3 | harddisk3

> The fourth hard disk.

n | net | network

> Boot over the ethernet network, using a PXE or BOOTP broadcast.

n | net | network | net1 | nic1 (HP Blade Only)

> Boot over the first ethernet network, using a PXE or BOOTP broadcast.

net2 | nic2 (HP Blade Only)

> Boot over the second ethernet network, using a PXE or BOOTP broadcast.

net3 | nic3 (HP Blade Only)

> Boot over the third ethernet network, using a PXE or BOOTP broadcast.

net3 | nic3 (HP Blade Only)

> Boot over the fourth ethernet network, using a PXE or BOOTP broadcast.

hfi

> Boot p775 nodes over the HFI network, using BOOTP broadcast.

iscsi

> Boot to an iSCSI disk over the network.

iscsicrit

> ??

cd | cdrom

> The CD or DVD drive.

usbflash | usb | flash

> A USB flash drive.

floppy

> The floppy drive.

none

> If it gets to this part of the sequence, do not boot. Can not be specified 1st, or before any real boot devices.

list | stat

> Display the current boot sequence.

EXAMPLES

1. Set blades 14-56 and 70-203 to try to boot first from the CD drive, then the floppy drive, then the network, and finally from the 1st hard disk:

```
rbootseq blade[14-56],blade[70-203] c,f,n,hd0
```

SEE ALSO

rsetboot(1)|rsetboot.1

rcons.1

Name

rcons - remotely accesses the serial console of a node

Synopsis

rcons *singlenode* [*conserver-host*] [**-f**] [**-s**]

rcons [**-h** | **--help** | **-v** | **--version**]

Description

rcons provides access to a single remote node serial console, using the out-of-band infrastructure for the node (e.g. BMC, Management Module, HMC, KVM, etc.). It uses the conserver open source package to provide one read-write and multiple read-only instances of the console, plus console logging.

If *conserver-host* is specified, the conserver daemon on that host will be contacted, instead of on the local host.

To exit the console session, enter: <ctrl><shift>e c .

Options

-f

> If another console for this node is already open in read-write mode, force that console into read-only (spy) mode, and open this console in read-write mode. If -f is not specified, this console will be put in spy mode if another console is already open in read-write mode. The -f flag can not be used with the -s flag.

-s

> Open the console in read-only (spy) mode, in this mode all the escape sequences work, but all other keyboard input is discarded. The -s flag can not be used with the -f flag.

-h | --help

> Print help.

-v | --version

> Print version.

Files

nodehm table - xCAT node hardware management table. See nodehm(5)|nodehm.5 for further details. This is used to determine the console access method.

Examples

rcons *node5*

See Also

wcons(1)|wcons.1

regnotif.1

NAME

regnotif - Registers a Perl module or a command that will get called when changes occur in the desired xCAT database tables.

SYNOPSIS

regnotif [-h| –help]

regnotif [-v| –version]

regnotif *filename tablename*[,tablename]... [-o|–operation actions]

DESCRIPTION

This command is used to register a Perl module or a command to the xCAT notification table. Once registered, the module or the command will get called when changes occur in the xCAT database tables indicated by tablename. The changes can be row addition, deletion and update which are specified by actions.

Parameters

filename is the path name of the Perl module or command to be registered. *tablename* is the name of the table that the user is interested in.

OPTIONS

-h | -help Display usage message.

-v | -version **** Command Version.

-V | -verbose Verbose output.

-o | -operation specifies the database table actions that the user is interested in. It is a comma separated list. 'a' for row addition, 'd' for row deletion and 'u' for row update.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To register a Perl module that gets invoked when rows get added or deleted. in the nodelist and the nodehm tables, enter:

```
regnotif /opt/xcat/lib/perl/xCAT_monitoring/mycode.pm nodelist,nodhm -o a,d
```

2. To register a command that gets invoked when rows get updated in the switch table, enter:

regnotif /usr/bin/mycmd switch -o u

FILES

/opt/xcat/bin/regnotif

SEE ALSO

unregnotif(1)|unregnotif.1

renergy.1

NAME

renergy - remote energy management tool

SYNOPSIS

renergy [-h | -****-help]

renergy [-v | -****-version]

Power 6 server specific :

renergy *noderange* [-V] { all | [savingstatus] [cappingstatus] [cappingmaxmin] [cappingvalue] [cappingsoftmin] [averageAC] [averageDC] [ambienttemp] [exhausttemp] [CPUspeed] [syssbpower] [sysIPLtime]]}

renergy *noderange* [-V] { savingstatus={on | off} | cappingstatus={on | off} | cappingwatt=watt | cappingperc=percentage }

Power 7 server specific :

renergy *noderange* [-V] { all | [savingstatus] [dsavingstatus] [cappingstatus] [cappingmaxmin] [cappingvalue] [cappingsoftmin] [averageAC] [averageDC] [ambienttemp] [exhausttemp] [CPUspeed] [syssbpower] [sysIPLtime] [fsavingstatus] [ffoMin] [ffoVmin] [ffoTurbo] [ffoNorm] [ffovalue]]}

renergy *noderange* [-V] { savingstatus={on | off} | dsavingstatus={on-norm | on-maxp | off} | fsavingstatus={on | off} | ffovalue=MHZ | cappingstatus={on | off} | cappingwatt=watt | cappingperc=percentage }

Power 8 server specific :

renergy *noderange* [-V] { all | [savingstatus] [dsavingstatus] [averageAC] [averageAChistory] [averageDC] [averageDChistory] [ambienttemp] [ambienttemphistory] [exhausttemp] [exhausttemphistory] [fanspeed] [fanspeedhistory] [CPUspeed] [CPUspeedhistory] [syssbpower] [sysIPLtime] [fsavingstatus] [ffoMin] [ffoVmin] [ffoTurbo] [ffoNorm] [ffovalue]]}

renergy *noderange* [-V] { savingstatus={on | off} | dsavingstatus={on-norm | on-maxp | off} | fsavingstatus={on | off} | ffovalue=MHZ }

NOTE: The setting operation for **Power 8** server is only supported for the server which is running in PowerVM mode. Do NOT run the setting for the server which is running in OPAL mode.

BladeCenter specific :

For Management Modules:

renergy *noderange* [-V] { all | pd1all | pd2all | [pd1status] [pd2status] [pd1policy] [pd2policy] [pd1powermodule1] [pd1powermodule2] [pd2powermodule1] [pd2powermodule2] [pd1avaiablepower] [pd2avaiablepower] [pd1reservedpower] [pd2reservedpower] [pd1remainpower] [pd2remainpower] [pd1inusedpower] [pd2inusedpower] [availableDC] [averageAC] [thermaloutput] [ambienttemp] [mmtemp] }

For a blade server nodes:

renergy *noderange* [-V] { all | [averageDC] [capability] [cappingvalue] [CPUspeed] [maxCPUspeed] [savingstatus] [dsavingstatus] }

renergy *noderange* [-V] { savingstatus={on | off} | dsavingstatus={on-norm | on-maxp | off} }

Flex specific :

For Flex Management Modules:

renergy *noderange* [-V] { all | [powerstatus] [powerpolicy] [powermodule] [avaiablepower] [reservedpower] [remainpower] [inusedpower] [availableDC] [averageAC] [thermaloutput] [ambienttemp] [mmtemp] }

For Flex node (power and x86):

renergy *noderange* [-V] { all | [averageDC] [capability] [cappingvalue] [cappingmaxmin] [cappingmax] [cappingmin] [cappingGmin] [CPUspeed] [maxCPUspeed] [savingstatus] [dsavingstatus] }

renergy *noderange* [-V] { cappingstatus={on | off} | cappingwatt=watt | cappingperc=percentage | savingstatus={on | off} | dsavingstatus={on-norm | on-maxp | off} }

iDataPlex specific :

renergy *noderange* [-V] [{ cappingmaxmin | cappingmax | cappingmin }] [cappingstatus] [cappingvalue] [relhistogram]

renergy *noderange* [-V] { cappingstatus={on | enable | off | disable} | {cappingwatt|cappingvalue}=watt }

DESCRIPTION

This **renergy** command can be used to manage the energy consumption of IBM servers which support IBM EnergyScale technology. Through this command, user can query and set the power saving and power capping status, and also can query the average consumed energy, the ambient and exhaust temperature, the processor frequency for a server.

renergy command supports IBM POWER6, POWER7 and POWER8 rack-mounted servers, BladeCenter management modules, blade servers, and iDataPlex servers. For *Power6* and *Power7* rack-mounted servers, the following specific hardware types are supported: *8203-E4A*, *8204-E8A*, *9125-F2A*, *8233-E8B*, *8236-E8C*. For *Power8* server, there's no hardware type restriction.

The parameter *noderange* needs to be specified for the **renergy** command to get the target servers. The *noderange* should be a list of CEC node names, blade management module node names or blade server node names. Lpar name is not acceptable here.

renergy command can accept multiple of energy attributes to query or one of energy attribute to set. If only the attribute name is specified, without the '=', **renergy**gets and displays the current value. Otherwise, if specifying the attribute with '=' like 'savingstatus=on', **renergy** will set the attribute savingstatus to value 'on'.

The attributes listed in the **SYNOPSIS** section are which ones can be handled by **renergy** command. But for each specific type of server, there are some attributes that are not supported. If user specifies an attribute which is not supported by a specific server, the return value of this attribute will be 'na'.

The supported attributes for each specific system p hardware type is listed as follows:

8203-E4A, **8204-E8A**

Supported attributes:

Query: savingstatus,cappingstatus,cappingmin,cappingmax, cappingvalue,cappingsoftmin,averageAC,averageDC,ambienttemp, exhausttemp,CPUspeed,syssbpower,sysIPLtime

Set: savingstatus,cappingstatus,cappingwatt,cappingperc

9125-F2A

Supported attributes:

Query: savingstatus,averageAC,ambienttemp,exhausttemp, CPUspeed

Set: savingstatus

8233-E8B, **8236-E8C**

Supported attributes:

Query: savingstatus,dsavingstatus,cappingstatus,cappingmin, cappingmax,cappingvalue,cappingsoftmin,averageAC,averageDC, ambienttemp,exhausttemp,CPUspeed,syssbpower,sysIPLtime

Set: savingstatus,dsavingstatus,cappingstatus,cappingwatt, cappingperc

9125-F2C, 9119-FHB

Supported attributes:

Query: savingstatus,dsavingstatus,cappingstatus,cappingmin, cappingmax,cappingvalue,cappingsoftmin,averageAC,averageDC, ambienttemp,exhausttemp,CPUspeed,syssbpower,sysIPLtime, fsavingstatus,ffoMin,ffoVmin,ffoTurbo,ffoNorm,ffovalue

Set: savingstatus,dsavingstatus,cappingstatus,cappingwatt, cappingperc,fsavingstatus,ffovalue

Non of Above

For the machine type which is not in the above list, the following attributes can be tried but not guaranteed:

Query: savingstatus,dsavingstatus,cappingstatus,cappingmin, cappingmax,,cappingvalue,cappingsoftmin,averageAC,averageDC, ambienttemp,exhausttemp,CPUspeed,syssbpower,sysIPLtime

Set: savingstatus,dsavingstatus,cappingstatus,cappingwatt, cappingperc

Note: For system P CEC nodes, each query operation for attribute CPUspeed, averageAC or averageDC needs about 30 seconds to complete. The query for others attributes will get response immediately.

PREREQUISITES

For the *Power6* and *Power7* nodes, the **renergy** command depends on the Energy Management Plugin **xCAT-pEnergy** to communicate with server. **xCAT-pEnergy** can be downloaded from the IBM web site: http://www.ibm.com/support/fixcentral/. (Other Software -> EM)

NOTE: *Power8* nodes don't need this specific energy management package.

For iDataPlex nodes, the **renergy** command depends on the Energy Management Plugin **xCAT-xEnergy** to communicate with server. This plugin must be requested from IBM.

(The support for BladeCenter energy management is built into base xCAT, so no additional plugins are needed for BladeCenter.)

OPTIONS

-h | --help

> Display the usage message.

-v | --version

> Display the version information.

-V

> Verbose output.

all

> Query all energy attributes which supported by the specific type of hardware.

> For *Power8* machines, will not display the attributes for historical records.

pd1all

Query all energy attributes of the power domain 1 for blade management module node.

pd2all

Query all energy attributes of the power domain 2 for blade management module node.

ambienttemp

Query the current ambient temperature. (Unit is centigrade)

ambienttemphistory

Query the historical records which were generated in last one hour for **ambienttemp**.

availableDC

Query the total DC power available for the entire blade center chassis.

averageAC

Query the average power consumed (Input). (Unit is watt)

Note: For 9125-F2A,9125-F2C server, the value of attribute averageAC is the aggregate for all of the servers in a rack.

Note: For Blade Center, the value of attribute averageAC is the total AC power being consumed by all modules

in the chassis. It also includes power consumed by the Chassis

Cooling Devices for BCH chassis.

averageAChistory

Query the historical records which were generated in last one hour for **averageAC**.

averageDC

Query the average power consumed (Output). (Unit is watt)

averageDChistory

Query the historical records which were generated in last one hour for **averageDC**.

capability

Query the Power Capabilities of the blade server.

staticPowerManagement: the module with the static worst case power values.

fixedPowermanagement: the module with the static power values but ability to throttle.

dynamicPowerManagement: the module with power meter capability, measurement enabled, but capping disabled.

dynamicPowerMeasurement1: the module with power meter capability, measurement enabled, phase 1 only

dynamicPowerMeasurement2: the module with power meter capability, measurement enabled, phase 2 or higher

dynamicPowerMeasurementWithPowerCapping: the module with power meter capability, both measurement and capping enabled, phase 2 or higher

cappingGmin

Query the Guaranteed Minimum power capping value in watts.

cappingmax

Query the Maximum of power capping value in watts.

cappingmaxmin

Query the Maximum and Minimum of power capping value in watts.

cappingmin

Query the Minimum of power capping value in watts.

cappingperc=percentage

Set the power capping value base on the percentage of the max-min of capping value which getting from *cappingmaxmim* attribute. The valid value must be from 0 to 100.

cappingsoftmin

Query the minimum value that can be assigned to power capping without guaranteed enforceability. (Unit is watt)

cappingstatus

Query the power capping status. The result should be 'on' or 'off'.

cappingstatus={on | off}

Set the power capping status. The value must be 'on' or 'off'. This is the switch to turn on or turn off the power capping function.

cappingvalue

Query the current power capping value. (Unit is watt)

cappingwatt=watt

Set the power capping value base on the watt unit.

If the 'watt' > maximum of *cappingmaxmin* or 'watt' < *cappingsoftmin*, the setting operation will be failed. If the 'watt' > *cappingsoftmin* and 'watt' < minimum of *cappingmaxmin*, the value can NOT be guaranteed.

CPUspeed

Query the effective processor frequency. (Unit is MHz)

CPUspeedhistory

Query the historical records which were generated in last one hour for **CPUspeed**

dsavingstatus

Query the dynamic power saving status. The result should be 'on-norm', 'on-maxp' or 'off'.

If turning on the dynamic power saving, the processor frequency and voltage will be dropped dynamically based on the core utilization. It supports two modes for turn on state:

on-norm - means normal, the processor frequency cannot exceed the nominal value;

on-maxp - means maximum performance, the processor frequency can exceed the nominal value.

dsavingstatus={on-norm | on-maxp | off}

Set the dynamic power saving. The value must be 'on-norm', 'on-maxp' or 'off'.

The dsavingstatus setting operation needs about 2 minutes to take effect. (The used time depends on the hardware type)

The **dsavingstatus** only can be turned on when the **savingstatus** is in turn off status.

exhausttemp

Query the current exhaust temperature. (Unit is centigrade)

exhausttemphistory

Query the historical records which were generated in last one hour for **exhausttemp**

fanspeed

Query the fan speed for all the fans which installed in this node. (Unit is RPM - Rotations Per Minute))

If there are multiple fans for a node, multiple lines will be output. And a fan name in bracket will be appended after **fanspped** attribute name.

fanspeedhistory

Query the historical records which were generated in last one hour for **fanspeed**.

ffoMin

Query the minimum cpu frequency which can be set for FFO. (Fixed Frequency Override)

ffoNorm

Query the maximum cpu frequency which can be set for FFO.

ffoTurbo

Query the advertised maximum cpu frequency (selling point).

ffoVmin

Query the minimum cpu frequency which can be set for dropping down the voltage to save power. That means when you drop the cpu frequency from the ffoVmin to ffoVmin, the voltage won't change, then there's no obvious power to be saved.

ffovalue

Query the current value of FFO.

ffovalue=MHZ

Set the current value of FFO. The valid value of ffovalue should be between the ffoMin and ffoNorm.

Note1: Due to the limitation of firmware, the frequency in the range 3501 MHz - 3807 MHz can NOT be set to ffovalue. This range may be changed in future.

Note2: The setting will take effect only when the fsavingstatus is in 'on' status. But you need to set the ffovalue to a valid value before enabling the fsavingstatus. (It's a limitation of the initial firmware and will be fixed in future.)

The ffovalue setting operation needs about 1 minute to take effect.

fsavingstatus

Query the status of FFO. The result should be 'on' or 'off'. 'on' - enable; 'off' - disable.

fsavingstatus={on | off}

Set the status of FFO. The value must be 'on' or 'off'.

'on' - enable. It will take effect only when the **ffovalue**has been set to a valid value.

'off' -disable. It will take effect immediately.

Note: See the Note2 of ffovalue=MHZ.

maxCPUspeed

Query the maximum processor frequency. (Unit is MHz)

mmtemp

Query the current temperature of management module. (Unit is centigrade)

pd1status | powerstatus

Query the status of power domain 1 for blade management module node.

Note: for the attribute without the leading 'pd1' which means there's only one power doamin in the chassis.

pd1policy | powerpolicy

Query the power management policy of power domain 1.

pd1powermodule1 | powermodule

Query the First Power Module capacity in power domain 1.

pd1powermodule2 | powermodule

Query the Second Power Module capacity in power domain 1.

pd1avaiablepower | avaiablepower

Query the total available power in power domain 1.

pd1reservedpower | reservedpower

Query the power that has been reserved for power domain 1.

pd1remainpower | remainpower

Query the remaining power available in power domain 1.

pd1inusedpower | inusedpower

Query the total power being used in power domain 1.

pd2status

Query the status of power domain 2 for blade management module node.

pd2policy

Query the power management policy of power domain 2.

pd2powermodule1

Query the First Power Module capacity in power domain 2.

pd2powermodule2

Query the Second Power Module capacity in power domain 2.

pd2avaiablepower

Query the total available power in power domain 2.

pd2reservedpower

Query the power that has been reserved for power domain 2.

pd2remainpower

Query the remaining power available in power domain 2.

pd2inusedpower

Query the total power being used in power domain 2.

relhistogram

Query histogram data for wattage information

savingstatus

Query the static power saving status. The result should be 'on' or 'off'. 'on' - enable; 'off' - disable.

savingstatus={on | off}

Set the static power saving. The value must be 'on' or 'off'.

If turning on the static power saving, the processor frequency and voltage will be dropped to a fixed value to save energy.

The savingstatus setting operation needs about 2 minutes to take effect. (The used time depends on the hardware type)

The **savingstatus** only can be turned on when the **dsavingstatus** is in turn off status.

sysIPLtime

Query the time used from FSP standby to OS standby. (Unit is Second)

syssbpower

Query the system power consumed prior to power on. (Unit is Watt)

thermaloutput

Query the thermal output (load) in BTUs per hour for the blade center chassis.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1

Query all attributes which CEC1,CEC2 supported.

renergy CEC1,CEC2 all

The output of the query operation:

```
CEC1: savingstatus: off
CEC1: dsavingstatus: off
CEC1: cappingstatus: off
CEC1: cappingmin: 1953 W
CEC1: cappingmax: 2358 W
CEC1: cappingvalue: 2000 W
CEC1: cappingsoftmin: 304 W
CEC1: averageAC: na
CEC1: averageDC: na
CEC1: ambienttemp: na
CEC1: exhausttemp: na
CEC1: CPUspeed: na
```

```
CEC1: syssbpower: 40 W
CEC1: sysIPLtime: 900 S
CEC2: savingstatus: off
CEC2: cappingstatus: off
CEC2: cappingmin: 955 W
CEC2: cappingmax: 1093 W
CEC2: cappingvalue: 1000 W
CEC2: cappingsoftmin: 226 W
CEC2: averageAC: 627 W
CEC2: averageDC: 531 W
CEC2: ambienttemp: 25 C
CEC2: exhausttemp: 40 C
CEC2: CPUspeed: 4695 MHz
```

2

Query the **fanspeed** attribute for Power8 CEC.

renergy CEC1 fanspeed

The output of the query operation:

```
CEC1: fanspeed (Fan U78CB.001.WZS00MA-A1 00002101): 5947 RPM
CEC1: fanspeed (Fan U78CB.001.WZS00MA-A2 00002103): 6081 RPM
CEC1: fanspeed (Fan U78CB.001.WZS00MA-A3 00002105): 6108 RPM
CEC1: fanspeed (Fan U78CB.001.WZS00MA-A4 00002107): 6000 RPM
CEC1: fanspeed (Fan U78CB.001.WZS00MA-A5 00002109): 6013 RPM
CEC1: fanspeed (Fan U78CB.001.WZS00MA-A6 0000210B): 6013 RPM
CEC1: fanspeed (Fan U78CB.001.WZS00MA-E1 0000210C): 4992 RPM
CEC1: fanspeed (Fan U78CB.001.WZS00MA-E2 0000210D): 5016 RPM
```

3

Query the historical records for the **CPUspeed** attribute. (Power8 CEC)

renergy CEC1 CPUspeedhistory

The output of the query operation:

```
CEC1: CPUspeedhistory: 2027 MHZ: 20141226042900
CEC1: CPUspeedhistory: 2027 MHZ: 20141226042930
CEC1: CPUspeedhistory: 2244 MHZ: 20141226043000
CEC1: CPUspeedhistory: 2393 MHZ: 20141226043030
CEC1: CPUspeedhistory: 2393 MHZ: 20141226043100
CEC1: CPUspeedhistory: 2393 MHZ: 20141226043130
CEC1: CPUspeedhistory: 2393 MHZ: 20141226043200
CEC1: CPUspeedhistory: 2393 MHZ: 20141226043230
CEC1: CPUspeedhistory: 2393 MHZ: 20141226043300
CEC1: CPUspeedhistory: 2393 MHZ: 20141226043330
...
```

4

Query all the attirbutes for management module node MM1. (For chassis)

renergy MM1 all

The output of the query operation:

```
    mm1: availableDC: 5880W
    mm1: frontpaneltmp: 18.00 Centigrade
```

```
    mm1: inusedAC: 2848W
    mm1: mmtmp: 28.00 Centigrade
    mm1: pd1avaiablepower: 2940W
    mm1: pd1inusedpower: 848W
    mm1: pd1policy: redundantWithoutPerformanceImpact
    mm1: pd1powermodule1: Bay 1: 2940W
    mm1: pd1powermodule2: Bay 2: 2940W
    mm1: pd1remainpower: 1269W
    mm1: pd1reservedpower: 1671W
    mm1: pd1status: 1 - Power domain status is good.
    mm1: pd2avaiablepower: 2940W
    mm1: pd2inusedpower: 1490W
    mm1: pd2policy: redundantWithoutPerformanceImpact
    mm1: pd2powermodule1: Bay 3: 2940W
    mm1: pd2powermodule2: Bay 4: 2940W
    mm1: pd2remainpower: 51W
    mm1: pd2reservedpower: 2889W
    mm1: pd2status: 2 - Warning: Power redundancy does not exist
in this power domain.
    mm1: thermaloutput: 9717.376000 BTU/hour
```

5

Query all the attirbutes for blade server node blade1.

renergy blade1 all

The output of the query operation:

```
blade1: CPUspeed: 4204MHZ
blade1: averageDC: 227W
blade1: capability: dynamicPowerMeasurement2
blade1: cappingvalue: 315W
blade1: dsavingstatus: off
blade1: maxCPUspeed: 4204MHZ
blade1: savingstatus: off
```

6

Query the attributes savingstatus, cappingstatus and CPUspeed for server CEC1.

renergy CEC1 savingstatus cappingstatus CPUspeed

The output of the query operation:

```
CEC1: savingstatus: off
CEC1: cappingstatus: on
CEC1: CPUspeed: 3621 MHz
```

7

Turn on the power saving function of CEC1.

renergy CEC1 savingstatus=on

The output of the setting operation:

```
CEC1: Set savingstatus succeeded.
CEC1: This setting may need some minutes to take effect.
```

8

Set the power capping value base on the percentage of the max-min capping value. Here, set it to 50%.

renergy CEC1 cappingperc=50

If the maximum capping value of the CEC1 is 850w, and the minimum capping value of the CEC1 is 782w, the Power Capping value will be set as ((850-782)*50% + 782) = 816w.

The output of the setting operation:

```
CEC1: Set cappingperc succeeded.
CEC1: cappingvalue: 816
```

REFERENCES

1

For more information on 'Power System Energy Management':

```
http://www-03.ibm.com/systems/power/software/energy/index.html
```

2

EnergyScale white paper for Power6:

```
http://www-03.ibm.com/systems/power/hardware/whitepapers/energyscale.html
```

3

EnergyScale white paper for Power7:

```
http://www-03.ibm.com/systems/power/hardware/whitepapers/energyscale7.html
```

FILES

/opt/xcat/bin/renergy

replaycons.1

NAME

replaycons - replay the console output for a node

SYNOPSIS

replaycons [*node*] [*bps*] [*tail_amount*]

replaycons [**-h** | **--help** | **-v** | **--version**]

DESCRIPTION

The **replaycons** command reads the console log stored by conserver for this node, and displays it in a way that simulates the original output of the console. Using the *bps* value, it will throttle the speed of the output play back. (The conserver logs are stored in /var/log/consoles.)

For now, replaycons must be run locally on the system on which the console log is stored. This is normally that management node, but in a hierarchical cluster will usually be the service node.

OPTIONS

bps]

> The display rate to use to play back the console output. Default is 19200.

tail_amount

> The place in the console log file to start play back, specified as the # of lines from the end.

-v|--version

> Command Version.

-h|--help

> Display usage message.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

1. To replay the console for node1 at the default rate, starting 2000 lines from the end:

```
replaycons 19200 2000
```

FILES

/opt/xcat/bin/replaycons

SEE ALSO

rcons(1)|rcons.1

restartxcatd.1

NAME

restartxcatd - Restart the xCAT daemon (xcatd).

SYNOPSIS

restartxcatd [[**-h** | **--help**] | [**-v** | **--version**] | [**-r** | **--reload**]] [**-V** | **--verbose**]

DESCRIPTION

The **restartxcatd** command restarts the xCAT daemon (xcatd).

Linux Specific

It will perform the xcatd *fast restart*. **The xcatd** *fast restart* **is a specific restart which has two advantages compares to the** *stop* **an**

1. The interval of xcatd out of service is very short.
2. The in processing request which initiated by old xcatd will not be stopped by force. The old xcatd will hand over the sockets to new xcatd, but old xcat will still be waiting for the in processing request to finish before the exit.

It does the same thing as 'service xcatd restart' on NON-systemd enabled Operating System like rh6.x and sles11.x. But for the systemd enabled Operating System like rh7 and sles12, the 'service xcatd restart' just do the *stop* and *start* instead of xcatd *fast restart*.

It's recommended to use **restartxcatd** command to restart xcatd on systemd enable system like rh7 and sles12 instead of 'service xcatd restart' or 'systemctl restart xcatd'.

AIX Specific

It runs 'stopsrc -s xcatd' to stop xcatd first if xcatd is active, then runs 'startsrc -s xcatd' to start xcatd.

If the xcatd subsystem was not created, **restartxcatd** will create it automatically.

OPTIONS

-h Display usage message.

-v Command Version.

-r On a Service Node, services will not be restarted.

-V Display the verbose messages.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To restart the xCAT daemon, enter:

restartxcatd

FILES

/opt/xcat/sbin/restartxcatd

restorexCATdb.1

NAME

restorexCATdb - restores the xCAT db tables .

SYNOPSIS

restorexCATdb [**-a**] [**-V**] [{**-p** | **--path**} *path*]

restorexCATdb [**-b**] [**-V**] [{**-t** | **--timestamp**} *timestamp*] [{**-p** | **--path**} *path*]

restorexCATdb [**-h** | **--help**] [**-v** | **--version**]

DESCRIPTION

If not using binary restore(-b), the restorexCATdb command restores the xCAT database tables from the *.csv files in directory given by the -p flag. The site table skiptables attribute can be set to a list of tables not to restore. It will not restore isnm_perf* tables. See man dumpxCATdb.

If using the binary restore option for DB2 or postgreSQL, the entire database is restored from the binary backup made with dumpxCATdb. The database will be restored using the database Utilities. For DB2, the timestamp of the correct DB2 backup file (-t) must be provided. All applications accessing the DB2 database must be stopped before you can use the binary restore options. See the xCAT DB2 document for more information. For postgreSQL, you do not have to stop the applications accessing the database and the complete path to the backup file, must be supplied on the -p flag.

OPTIONS

-h Display usage message.

-v Command Version.

-V Verbose.

-a All,without this flag the eventlog and auditlog will be skipped. These tables are skipped by default because restoring will generate new indexes

-b Restore from the binary image.

-p Path to the directory containing the database restore files. If restoring from the binary image (-b) and using postgeSQL, then this is the complete path to the restore file that was created with dumpxCATdb -b.

-t Use with the -b flag to designate the timestamp of the binary image to use to restore for DB2.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To restore the xCAT database from the /dbbackup/db directory, enter:

restorexCATdb -p /dbbackup/db

2. To restore the xCAT database including auditlog and eventlog from the /dbbackup/db directory, enter:

restorexCATdb -a -p /dbbackup/db

3. To restore the xCAT DB2 database from the binary image with timestamp 20111130130239 enter:

restorexCATdb -b -t 20111130130239 -p /dbbackup/db

4. To restore the xCAT postgreSQL database from the binary image file pgbackup.20553 created by dumpxCATdb enter:

restorexCATdb -b -p /dbbackup/db/pgbackup.20553

FILES

/opt/xcat/sbin/restorexCATdb

SEE ALSO

dumpxCATdb(1)|dumpxCATdb.1

reventlog.1

Name

reventlog - retrieve or clear remote hardware event logs

Synopsis

reventlog *noderange* {*number-of-entries [-s]*|**all [-s] | clear**}

reventlog [**-h | --help | -v | --version**]

Description

reventlog can display any number of remote hardware event log entries or clear them for a range of nodes. Hardware event logs are stored on each servers service processor.

Options

number-of-entries

Retrieve the specified number of entries from the nodes' service processors.

all

Retrieve all entries.

-s

To sort the entries from latest (always the last entry in event DB) to oldest (always the first entry in event DB). If **number-of-entries** specified, the latest **number-of-entries** events will be output in the order of latest to oldest.

clear

Clear event logs.

-h | --help

Print help.

-v | --version

Print version.

Examples

reventlog *node4,node5 5*

```
node4: SERVPROC  I 09/06/00 15:23:33 Remote Login Successful User ID = USERID[00]
node4: SERVPROC I 09/06/00 15:23:32 System spn1 started a RS485 connection with us[00]
node4: SERVPROC I 09/06/00 15:22:35 RS485 connection to system spn1 has ended[00]
node4: SERVPROC I 09/06/00 15:22:32 Remote Login Successful User  ID  = USERID[00]
node4: SERVPROC I 09/06/00 15:22:31 System spn1 started a RS485 connection with us[00]
node5: SERVPROC I 09/06/00 15:22:32 Remote Login Successful User  ID  = USERID[00]
node5: SERVPROC I 09/06/00 15:22:31 System spn1 started a RS485 connection with us[00]
node5: SERVPROC I 09/06/00 15:21:34 RS485 connection to system spn1 has ended[00]
node5:  SERVPROC  I 09/06/00 15:21:30 Remote Login Successful User ID = USERID[00]
node5: SERVPROC I 09/06/00 15:21:29 System spn1 started a RS485 connection with us[00]
```

reventlog *node4,node5 clear*

```
node4: clear
node5: clear
```

SEE ALSO

rpower(1)|rpower.1, monstart(1)|monstart.1

rflash.1

Name

rflash - Performs Licensed Internal Code (LIC) update support for HMC-attached POWER5 and POWER6 Systems, and POWER7 systems using Direct FSP management. rflash is also able to update firmware for NextScale Fan Power Controllers (FPC).

Synopsis

rflash [**-h** | **--help** | **-v** | **--version**]

PPC (with HMC) specific:

rflash *noderange* **-p** *directory* {**--activate concurrent** | **disruptive**} [**-V** | **--verbose**]

rflash *noderange* {**--commit** | **--recover**} [**-V** | **--verbose**]

PPC (without HMC, using Direct FSP Management) specific:

rflash *noderange* **-p** *directory* **--activate disruptive** | **deferred** [**-d** *data_directory*]

rflash *noderange* {**--commit** | **--recover**}

NeXtScale FPC specific:

rflash *noderange http directory*

Description

rflash The **rflash** command initiates Firmware updates on supported xCAT nodes. Licensed Internal Code (also known as microcode) updates are performed on supported HMC-attached POWER5 and POWER6 pSeries nodes, and POWER7 systems using Direct FSP management.

The command scans the specified directory structure for Firmware update package files applicable to the given nodes and components. And then it will **automatically** select the **latest** version for the upgrade. The firmware update files include the Microcode update package and associated XML file. They can be downloaded from the IBM Web site: *http://www-933.ibm.com/support/fixcentral/*.

The POWER5 and POWER6 systems contain several components that use Licensed Internal Code. The **rflash** command supports two of these components: the managed system (also known as the Central Electronics Complex, or CEC) and the power subsystem (also known as the Bulk Power Assembly (BPA) or Bulk Power Controller (BPC)). Some POWER5 managed systems can be attached to a power subsystem. These power subsystems can support multiple managed systems. When the **rflash** command is invoked, xCAT will determine the managed system or power subsystem associated with that CEC and perform the update.

The **noderange** can be an CEC or CEC list, a Lpar or Lpar list and a Frame or Frame list. But CEC (or Lpar) and Frame **can't** be used at the same time. When the *noderange* is an CEC or CEC list, **rflash** will upgrade the firmware of the CEC or CECs in the cec list. If *noderange* is a Lpar or Lpar list, **rflash** will update Licensed Internal Code (LIC) on HMC-attached POWER5 and POWER6 pSeries nodes, and POWER7 systems using Direct FSP management. If *noderange* is a Frame or Frame list, **rflash** will update Licensed Internal Code (LIC) of the power subsystem on HMC-attached POWER5 and POWER6 pSeries nodes. The *noderange* can also be the specified node groups. You

can specify a comma or space-separated list of node group ranges. See the *noderange* man page for detailed usage information.

The command will update firmware for NeXtScale FPC when given an FPC node and the http information needed to access the firmware.

PPC (with HMC) specific:

The **rflash** command uses the **xdsh** command to connect to the HMC controlling the given managed system and perform the updates. Before run **rflash**, please use **rspconfig** to check if the related HMC ssh is enabled. If enable a HMC ssh connection, please use **rspconfig** comamnd.

Warning! This command may take considerable time to complete, depending on the number of systems being updated and the workload on the target HMC. In particular, power subsystem updates may take an hour or more if there are many attached managed systems.

Depending on the Licensed Internal Code update that is installed, the affected HMC-attached POWER5 and POWER6 systems may need to be recycled. The **--activate** flag determines how the affected systems activate the new code. The concurrent option activates code updates that do not require a system recycle (known as a "concurrent update"). If this option is given with an update that requires a system recycle (known as a "disruptive update"), a message will be returned, and no activation will be performed. The disruptive option will cause any affected systems that are powered on to be powered down before installing and activating the update. Once the update is complete, the command will attempt to power on any affected systems that it powered down. Those systems that were powered down when the command was issued will remain powered down when the update is complete.

The flash chip of a POWER5 and POWER6 managed system or power subsystem stores firmware in two locations, referred to as the temporary side and the permanent side. By default, most POWER5 and POWER6 systems boot from the temporary side of the flash. When the **rflash** command updates code, the current contents of the temporary side are written to the permanent side, and the new code is written to the temporary side. The new code is then activated. Therefore, the two sides of the flash will contain different levels of code when the update has completed.

The **--commit** flag is used to write the contents of the temporary side of the flash to the permanent side. This flag should be used after updating code and verifying correct system operation. The **--recover** flag is used to write the permanent side of the flash chip back to the temporary side. This flag should be used to recover from a corrupt flash operation, so that the previously running code can be restored.

NOTE: When the **--commit** or **--recover** two flags is used, the noderange **cannot** be BPA. It only **can** be CEC or LPAR ,and will take effect for **both** managed systems and power subsystems.

xCAT recommends that you shutdown your Operating System images and power off your managed systems before applying disruptive updates to managed systems or power subsystems.

Any previously activated code on the affected systems will be automatically accepted into permanent flash by this procedure.

IMPORTANT! If the power subsystem is recycled, all of its attached managed systems will be recycled.

If it outputs **"Timeout waiting for prompt"** during the upgrade, please set the **"ppctimeout"** larger in the **site** table. After the upgrade, remeber to change it back. If run the **"rflash"** command on an AIX management node, need to make sure the value of **"useSSHonAIX"** is **"yes"** in the site table.

PPC (using Direct FSP Management) specific:

In currently Direct FSP/BPA Management, our **rflash** doesn't support **concurrent** value of **--activate** flag, and supports **disruptive** and **deferred**. The **disruptive** option will cause any affected systems that are powered on to be powered down before installing and activating the update. So we require that the systems should be powered off before do the firmware update.

The deferred option will load the new firmware into the T (temp) side, but will not activate it like the disruptive firmware. The o
   XCAT_Power_775_Hardware_Management

In Direct FSP/BPA Management, there is -d <data_directory> option. The default value is /tmp. When do firmware
update, rflash will put some related data from rpm packages in <data_directory> directory, so the execution of rflash
will require available disk space in <data_directory> for the command to properly execute:

For one GFW rpm package and one power code rpm package , if the GFW rpm package size is gfw_rpmsize, and the Power co
   1.5*gfw_rpmsize + 1.5*power_rpmsize

For Power 775, the rflash command takes effect on the primary and secondary FSPs or BPAs almost in parallel.

For more details about the Firmware Update using Direct FSP/BPA Management, refer to:
   XCAT_Power_775_Hardware_Management#Updating_the_BPA_and_FSP_firmware_using_xCAT_DFM

NeXtScale FPC specific:

The command will update firmware for NeXtScale FPC when given an FPC node and the http information needed to access the
   XCAT_NeXtScale_Clusters

Options

-h|--help

   Writes the command's usage statement to standard output.

-p directory

   Specifies the directory where the packages are located.

-d data_directory

   Specifies the directory where the raw data from rpm packages for each CEC/Frame are located. The
   default directory is /tmp. The option is only used in Direct FSP/BPA Management.

--activate concurrent | disruptive

   Must be specified to activate the new Licensed Internal Code. The "disruptive" option will cause the target
   systems to be recycled. Without this flag, LIC updates will be installed only, not activated.

--commit

   Used to commit the flash image in the temporary side of the chip to the permanent side for both managed
   systems and power subsystems.

--recover

   Used to recover the flash image in the permanent side of the chip to the temporary side for both managed
   systems and power subsystems.

-v|--version

   Displays the command's version.

-V|--verbose

   Verbose output.

Exit Status

0 The command completed successfully.

1 An error has occurred.

Examples

1

> To update only the power subsystem attached to a single HMC-attached pSeries CEC(cec_name), and recycle the power subsystem and all attached managed systems when the update is complete, and the Microcode update package and associated XML file are in /tmp/fw, enter:

```
rflash cec_name -p /tmp/fw --activate disruptive
```

2

> To update only the power subsystem attached to a single HMC-attached pSeries node, and recycle the power subsystem and all attached managed systems when the update is complete, and the Microcode update package and associated XML file are in /tmp/fw, enter:

```
rflash bpa_name -p /tmp/fw --activate disruptive
```

3

> To commit a firmware update to permanent flash for both managed system and the related power subsystems, enter:

```
rflash cec_name --commit
```

4

> To update the firmware on a NeXtScale FPC specify the FPC node name and the HTTP location of the file including the xCAT MN IP address and the directory on the xCAT MN containing the firmware as follows:

```
rflash fpc01 http://10.1.147.169/install/firmware/fhet17a/ibm_fw_fpc_fhet17a-
↪2.02_anyos_noarch.rom
```

Location

/opt/xcat/bin/rflash

NOTES

This command is part of the xCAT software product.

SEE ALSO

rinv(1)|rinv.1, rspconfig(1)|rspconfig.1

rinv.1

Name

rinv - Remote hardware inventory

Synopsis

rinv [**-h** | **--help** | **-v** | **--version**]

BMC/MPA specific:

rinv *noderange* {**pci** | **model** | **serial** | **asset** | **vpd** | **mprom** | **deviceid** | **guid** | **firm** | **diag** | **bios** | **mparom** | **mac** | **all**}

PPC (with HMC) specific:

rinv *noderange* {**bus** | **config** | **serial** | **model** | **firm** | **all**}

PPC (using Direct FSP Management) specific:

rinv *noderange* {**firm**}

rinv *noderange* {**deconfig** [**-x**]}

Blade specific:

rinv *noderange* {**mtm** | **serial** | **mac** | **bios** | **diag** | **mprom** | **mparom** | **firm** | **all**}

VMware specific:

rinv *noderange* [**-t**]

zVM specific:

rinv *noderange* [**config** | **all**]

rinv *noderange* [**--diskpoolspace**]

rinv *noderange* [**--diskpool** *pool space*]

rinv *noderange* [**--fcpdevices** *state details*]

rinv *noderange* [**--diskpoolnames**]

rinv *noderange* [**--networknames**]

rinv *noderange* [**--network** *name*]

rinv *noderange* [**--ssi**]

rinv *noderange* [**--smapilevel**]

rinv *noderange* [**--wwpns** *fcp_channel*]

rinv *noderange* [**--zfcppool** *pool space*]

rinv *noderange* [**--zfcppoolnames**]

Description

rinv retrieves hardware configuration information from the on-board Service Processor for a single or range of nodes and groups.

Calling **rinv** for VMware will display the UUID/GUID, nuumber of CPUs, amount of memory, the MAC address and a list of Hard disks. The output for each Hard disk includes the label, size and backing file location.

Options

pci

> Retrieves PCI bus information.

bus

> List all buses for each I/O slot.

config

> Retrieves number of processors, speed, total memory, and DIMM locations.

model

> Retrieves model number.

serial

> Retrieves serial number.

firm

> Retrieves firmware versions.

deconfig

> Retrieves deconfigured resources. Deconfigured resources are hw components (cpus, memory, etc.) that have failed so the firmware has automatically turned those components off. This option is only capable of listing some of the deconfigured resources and should not be the only method used to check the hardware status.

-x

> To output the raw information of deconfigured resources for CEC.

asset

> Retrieves asset tag. Usually it's the MAC address of eth0.

vpd

> Same as specifying model, serial, deviceid, and mprom.

diag

> Diagnostics information of firmware.

mprom

Retrieves mprom firmware level

deviceid

Retrieves device identification. Usually device, manufacturing and product ids.

guid

Retrieves the global unique identifier

all

All of the above.

-h | --help

Print help.

-v | --version

Print version.

-t

Set the values in the vm table to what vCenter has for the indicated nodes.

zVM specific :

--diskpoolspace

Calculates the total size of every known storage pool.

--diskpool *pool space*

Lists the storage devices (ECKD and FBA) contained in a disk pool. Space can be: all, free, or used.

--fcpdevices *state details*

Lists the FCP device channels that are active, free, or offline. State can be: active, free, or offline.

--diskpoolnames

Lists the known disk pool names.

--networknames

Lists the known network names.

--network *name*

Shows the configuration of a given network device.

--ssi

Obtain the SSI and system status.

--smapilevel

Obtain the SMAPI level installed on the z/VM system.

--wwpns *fcp_channel*

Query a given FCP device channel on a z/VM system and return a list of WWPNs.

--zfcppool *pool space*

List the SCSI/FCP devices contained in a zFCP pool. Space can be: free or used.

--zfcppoolnames

List the known zFCP pool names.

Examples

*

To retrieve all information available from blade node4, enter:

```
rinv node5 all

node5: Machine Type/Model 865431Z
node5: Serial Number 23C5030
node5: Asset Tag 00:06:29:1F:01:1A
node5: PCI Information
node5:  Bus  VendID  DevID   RevID  Description              Slot Pass/Fail
node5:  0    1166    0009    06     Host Bridge              0    PASS
node5:  0    1166    0009    06     Host Bridge              0    PASS
node5:  0    5333    8A22    04     VGA Compatible Controller0   PASS
node5:  0    8086    1229    08     Ethernet Controller      0    PASS
node5:  0    8086    1229    08     Ethernet Controller      0    PASS
node5:  0    1166    0200    50     ISA Bridge               0    PASS
node5:  0    1166    0211    00     IDE Controller           0    PASS
node5:  0    1166    0220    04     Universal Serial Bus     0    PASS
node5:  1    9005    008F    02     SCSI Bus Controller      0    PASS
node5:  1    14C1    8043    03     Unknown Device Type      2    PASS
node5: Machine Configuration Info
node5: Number of Processors:
node5: Processor Speed: 866 MHz
node5: Total Memory:    512 MB
node5: Memory DIMM locations:  Slot(s)  3  4
```

*

To output the raw information of deconfigured resources for CEC cec01, enter:

```
rinv cec01 deconfig -x

cec01:
<SYSTEM>
<System_type>IH</System_type>
<NODE>
<Location_code>U78A9.001.0123456-P1</Location_code>
<RID>800</RID>
</NODE>
</SYSTEM>
```

*

To retrieve 'config' information from the HMC-managed LPAR node3, enter:

```
rinv node3 config

node5: Machine Configuration Info
node5: Number of Processors: 1
node5: Total Memory (MB): 1024
```

*

To retrieve information about a VMware node vm1, enter:

```
rinv vm1
vm1: UUID/GUID: 42198f65-d579-fb26-8de7-3ae49e1790a7
vm1: CPUs: 1
vm1: Memory: 1536 MB
vm1: Network adapter 1: 36:1b:c2:6e:04:02
vm1: Hard disk 1 (d0): 9000 MB @ [nfs_192.168.68.21_vol_rc1storage_vmware]␣
↪vm1_3/vm1.vmdk
vm1: Hard disk 2 (d4): 64000 MB @ [nfs_192.168.68.21_vol_rc1storage_vmware]␣
↪vm1_3/vm1_5.vmdk
```

zVM specific :

*

To list the defined network names available for a given node:

```
rinv pokdev61 --getnetworknames
```

Output is similar to:

```
pokdev61: LAN:QDIO SYSTEM GLAN1
pokdev61: LAN:HIPERS SYSTEM GLAN2
pokdev61: LAN:QDIO SYSTEM GLAN3
pokdev61: VSWITCH SYSTEM VLANTST1
pokdev61: VSWITCH SYSTEM VLANTST2
pokdev61: VSWITCH SYSTEM VSW1
pokdev61: VSWITCH SYSTEM VSW2
pokdev61: VSWITCH SYSTEM VSW3
```

*

To list the configuration for a given network:

```
rinv pokdev61 --getnetwork GLAN1
```

Output is similar to:

```
pokdev61: LAN SYSTEM GLAN1          Type: QDIO     Connected: 1     Maxconn:␣
↪INFINITE
pokdev61:   PERSISTENT  UNRESTRICTED  IP                            Accounting:␣
↪OFF
pokdev61:   IPTimeout: 5                   MAC Protection: Unspecified
pokdev61:   Isolation Status: OFF
```

*

To list the disk pool names available:

```
rinv pokdev61 --diskpoolnames
```

Output is similar to:

```
pokdev61: POOL1
pokdev61: POOL2
pokdev61: POOL3
```

*

List the configuration for a given disk pool:

```
rinv pokdev61 --diskpool POOL1 free
```

Output is similar to:

```
pokdev61: #VolID DevType StartAddr Size
pokdev61: EMC2C4 3390-09 0001 10016
pokdev61: EMC2C5 3390-09 0001 10016
```

*

List the known zFCP pool names.

```
rinv pokdev61 --zfcppoolnames
```

Output is similar to:

```
pokdev61: zfcp1
pokdev61: zfcp2
pokdev61: zfcp3
```

*

List the SCSI/FCP devices contained in a given zFCP pool:

```
rinv pokdev61 --zfcppool zfcp1
```

Output is similar to:

```
pokdev61: #status,wwpn,lun,size,range,owner,channel,tag
pokdev61: used,500512345678c411,4014412100000000,2g,3B40-3B7F,ihost13,3b77,
pokdev61: used,500512345678c411,4014412200000000,8192M,3B40-3B7F,ihost13,
→3b77,replace_root_device
pokdev61: free,500512345678c411,4014412300000000,8g,3B40-3B7F,,,
pokdev61: free,5005123456789411,4014412400000000,2g,3B40-3B7F,,,
pokdev61: free,5005123456789411;5005123456789411,4014412600000000,2G,3B40-
→3B7F,,,
```

SEE ALSO

rpower(1)|rpower.1

rmdef.1

NAME

rmdef - Use this command to remove xCAT data object definitions.

SYNOPSIS

rmdef [**-h** | **--help**] [**-t** *object-types*]

rmdef [**-V** | **--verbose**] [**-a** | **--all**] [**-t** *object-types*] [**-o** *object-names*] [**-f** | **--force**] [*noderange*]

DESCRIPTION

This command is used to remove xCAT object definitions that are stored in the xCAT database.

OPTIONS

-a|--all

> Clear the whole xCAT database. A backup of the xCAT definitions should be saved before using this option. Once all the data is removed the xCAT daemon will no longer work. Most xCAT commands will fail. In order to use xCAT commands again, you have two options. You can restore your database from your backup by switching to bypass mode, and running the restorexCATdb command. You switch to bypass mode by setting the XCATBYPASS environmant variable. (ex. "export XCATBYPASS=yes") A second option is to run xcatconfig -d. This will restore the initial setup of the database as when xCAT was initially installed. You can then restart xcatd and run xCAT commands.

-f|--force

> Use this with the all option as an extra indicator that ALL definitions are to be removed.

-h|--help

> Display a usage message.

noderange

> A set of comma delimited node names and/or group names. See the "noderange" man page for details on supported formats.

-o *object-names*

> A set of comma delimited object names.

-t *object-types*

> A set of comma delimited object types.

-V|--verbose

> Verbose mode.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To remove a range of node definitions.

```
rmdef -t node node1-node4
```

2. To remove all node definitions for the nodes contained in the group bpcnodes.

```
rmdef -t node -o bpcnodes
```

3. To remove the group called bpcnodes.

```
rmdef -t group -o bpcnodes
```

(This will also update the values of the "groups" attribute of the member nodes.)

FILES

$XCATROOT/bin/rmdef

(The XCATROOT environment variable is set when xCAT is installed. The default value is "/opt/xcat".)

NOTES

This command is part of the xCAT software product.

SEE ALSO

mkdef(1)|mkdef.1, lsdef(1)|lsdef.1, chdef(1)|chdef.1, xcatstanzafile(5)|xcatstanzafile.5

rmdsklsnode.1

NAME

rmdsklsnode - Use this xCAT command to remove AIX/NIM diskless machine definitions.

SYNOPSIS

rmdsklsnode [-h | --help]

rmdsklsnode [-V|--verbose] [-f|--force] [-r|--remdef] [-i image_name] [-p|--primarySN] [-b|--backupSN] noderange

DESCRIPTION

Use this command to remove all NIM client machine definitions that were created for the specified xCAT nodes.

The xCAT node definitions will not be removed. Use the xCAT **rmdef** command to remove xCAT node definitions.

If you are using xCAT service nodes the **rmdsklsnode** command will automatically determine the correct server(s) for the node and remove the NIM definitions on that server(s).

If the node you are trying to remove is currently running the **rmdsklsnode** command will not remove the definitions. You can use the "-f" option to shut down the node and remove the definition.

Removing alternate NIM client definitions

If you used the "-n" option when you created the NIM client definitions with the **mkdsklsnode** command then the NIM client machine names would be a combination of the xCAT node name and the osimage name used to initialize the NIM machine. To remove these definitions you must provide the name of the osimage that was used using the "-i" option.

In most cases you would most likely want to remove the old client definitions without disturbing the nodes that you just booted with the new alternate client definition. The **rmdsklsnode -r** option can be used to remove the old alternate client defintions without stopping the running node.

However, if you have NIM dump resources assign to your nodes be aware that when the old NIM alternate client definitions are removed it will leave the nodes unable to produce a system dump. This is a current limitation in the NIM support for alternate client definitions. For this reason it is recommended that you wait to do this cleanup until right before you do your next upgrade.

OPTIONS

-f |--force

> Use the force option to stop and remove running nodes. This handles the situation where a NIM machine definition indicates that a node is still running even though it is not.

-b |--backupSN

> When using backup service nodes only update the backup. The default is to updat e both the primary and backup service nodes.

-h |--help

> Display usage message.

-i image_name

> The name of an xCAT image definition.

noderange

> A set of comma delimited node names and/or group names. See the "noderange" man page for details on additional supported formats.

-p|--primarySN

> When using backup service nodes only update the primary. The default is to upda te both the primary and backup service nodes.

-r|--remdef

> Use this option to reset, deallocate, and remove NIM client definitions. This option will not attempt to shut down running nodes. This option should be used when remove alternate NIM client definitions that were created using **mkdsklsnode -n**.

-V |--verbose

> Verbose mode.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

1. Remove the NIM client definition for the xCAT node named "node01". Give verbose output.

rmdsklsnode -V node01

2) Remove the NIM client definitions for all the xCAT nodes in the group "aixnod es". Attempt to shut down the nodes if they are running.

rmdsklsnode -f aixnodes

3. Remove the NIM client machine definition for xCAT node "node02" that was created with the **mkdsklsnode -n** option and the image "AIXdskls". (i.e. NIM client machine name "node02_AIXdskls".)

rmdsklsnode -i AIXdskls node02

This assume that node02 is not currently running.

4. Remove the old alternate client definition "node27_olddskls".

rmdsklsnode -r -i olddskls node27

Assuming the node was booted using an new alternate NIM client definition then this will leave the node running.

FILES

/opt/xcat/bin/rmdsklsnode

NOTES

This command is part of the xCAT software product.

SEE ALSO

mkdsklsnode(1)|mkdsklsnode.1

rmflexnode.1

NAME

rmflexnode - Delete a flexible node.

SYNOPSIS

rmflexnode [-h | -****-help]

rmflexnode [-v | -****-version]

rmflexnode *noderange*

DESCRIPTION

Delete a flexible node which created by the **mkflexnode** command.

The **rmflexnode** command will delete the **Partition** which the slots in *id* attribute assigned to.

The action of deleting flexible node will impact the hardware status. Before deleting it, the blades in the slot range should be in **power off** state.

After the deleting, use the **lsflexnode** to check the status of the node.

The *noderange* only can be a blade node.

OPTIONS

-h | --help

> Display the usage message.

-v | --version

> Display the version information.

EXAMPLES

1

> Delete a flexible node base on the xCAT node blade1.
>
> The blade1 should belong to a complex, the *id* attribute should be set correctly and all the slots should be in **power off** state.

```
rmflexnode blade1
```

FILES

/opt/xcat/bin/rmflexnode

SEE ALSO

lsflexnode(1)|lsflexnode.1, mkflexnode(1)|mkflexnode.1

rmhwconn.1

NAME

rmhwconn - Use this command to remove connections from CEC and Frame nodes to HMC nodes.

SYNOPSIS

rmhwconn [**-h**| **--help**]

rmhwconn [**-v**| **--version**]

PPC (with HMC) specific:

rmhwconn [**-V**| **--verbose**] *noderange*

PPC (without HMC, using FSPAPI) specific:

rmhwconn *noderange* **-T** *tooltype*

PPC (use HMC as SFP) specific:

rmhwconn -s

DESCRIPTION

For PPC (with HMC) specific:

This command is used to disconnect CEC and Frame nodes from HMC nodes, according to the connection information defined in ppc talbe in xCAT DB.

Note: If a CEC belongs to a frame with a BPA installed, this CEC cannot be disconnected individually. Instead, the whole frame should be disconnected.

For PPC (without HMC, using FSPAPI) specific:

It's used to disconnection CEC and Frame nodes from hardware server.

For PPC (use HMC as SFP) specific:

It is used to disconnect Frame nodes from HMC nodes.

OPTIONS

-h|--help

  Display usage message.

-V|--verbose

  Verbose output.

-T

  The tooltype is used to communicate to the CEC/Frame. The value could be **lpar** or **fnm**. The tooltype value **lpar** is for xCAT and **fnm** is for CNM.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To disconnect all CEC nodes in node group cec from their HMC nodes:

```
rmhwconn cec
```

2. To remove the connection for Frame node frame1:

```
rmhwconn frame1
```

3. To disconnect all CEC nodes in node group cec from their related hardware serveri, using lpar tooltype:

```
rmhwconn cec -T lpar
```

FILES

$XCATROOT/bin/rmhwconn

(The XCATROOT environment variable is set when xCAT is installed. The default value is "/opt/xcat".)

NOTES

This command is part of the xCAT software product.

SEE ALSO

lshwconn(1)|lshwconn.1, mkhwconn(1)|mkhwconn.1

rmhypervisor.1

NAME

rmhypervisor - Remove the virtualization hosts.

SYNOPSIS

RHEV specific :

rmhypervisor *noderange* [**-f**]

DESCRIPTION

The **rmhypervisor** command can be used to remove the virtualization host.

OPTIONS

-f

> If **-f** is specified, the host will be deactivated to maintenance before the removing.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To remove the host 'host1', enter:

```
rmhypervisor host1
```

FILES

/opt/xcat/bin/rmhypervisor

rmigrate.1

Name

rmigrate - Execute migration of a guest VM between hosts/hypervisors

Synopsis

rmigrate *noderange target_host*

For zVM:

rmigrate *noderange* [**destination=***target_host*] [**action=***action*] [**force=***force*] [**immediate=***yes_no*] [**max_total=***total*] [**max_quiesce=***quiesce*]

Description

rmigrate requests that a guest VM be moved from the current entity hosting it to another. It requests a live migration be done, if possible.

For zVM:

rmigrate migrates a VM from one z/VM member to another in an SSI cluster (only in z/VM 6.2).

OPTIONS

zVM specific:

destination= The name of the destination z/VM system to which the specified virtual machine will be relocated.

action= It can be: (MOVE) initiate a VMRELOCATE MOVE of the VM, (TEST) determine if VM is eligible to be relocated, or (CANCEL) stop the relocation of VM.

force= It can be: (ARCHITECTURE) attempt relocation even though hardware architecture facilities or CP features are not available on destination system, (DOMAIN) attempt relocation even though VM would be moved outside of its domain, or (STORAGE) relocation should proceed even if CP determines that there are insufficient storage resources on destination system.

immediate= It can be: (YES) VMRELOCATE command will do one early pass through virtual machine storage and then go directly to the quiesce stage, or (NO) specifies immediate processing.

max_total= The maximum wait time for relocation to complete.

max_quiesce= The maximum quiesce time a VM may be stopped during a relocation attempt.

Files

vm table - Table governing VM paramaters. See vm(5)|vm.5 for further details. This is used to determine the current host to migrate from.

Examples

rmigrate *v1 n2*

zVM specific:

rmigrate *ihost123* destination=*pokdev62*

rmimage.1

NAME

rmimage - Removes the Linux stateless or statelite image from the file system.

SYNOPSIS

rmimage [-h | –help]

rmimage [-V | –verbose] -o <OS> -a <architecture> -p <profile>

rmimage [-V | –verbose] imagename [–xcatdef]

DESCRIPTION

Removes the Linux stateless or statelite image from the file system. The install dir is setup by using "installdir" attribute set in the site table.

If *imagename* is specified, this command uses the information in the *imagename*to calculate the image root directory; otherwise, this command uses the operating system name, architecture and profile name to calculate the image root directory.

The osimage definition will not be removed from the xCAT tables by default, specifying the flag –xcatdef will remove the osimage definition, or you can use rmdef -t osimage to remove the osimage definition.

The statelite image files on the diskful service nodes will not be removed, remove the image files on the service nodes manually if necessary, for example, use command "rsync -az –delete /install <sn>:/" to remove the image files on the service nodes, where the <sn> is the hostname of the service node.

Parameters

imagename specifies the name of an os image definition to be used. The specification for the image is stored in the *osimage* table and *linuximage* table.

OPTIONS

-h | --help Display usage message.

-V | --verbose Verbose mode.

-o Operating system (fedora9,rhels5.5,sles11,etc)

-p Profile (compute,service,etc)

-a Architecture (ppc64,x86_64,etc)

--xcatdef Remove the xCAT osimage definition

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To remove a fedora9 stateless image for a compute node architecture x86_64, enter:

rmimage fedora9-x86_64-netboot-compute

or

rmimage -o fedora9 -p compute -a x86_64

2. To remove a rhels5.5 statelite image for a compute node architecture ppc64 and the osimage definition, enter:

rmimage rhels5.5-ppc64-statelite-compute –xcatdef

or

rmimage -o rhels5.5 -p compute -a ppc64 –xcatdef

FILES

/opt/xcat/sbin/rmimage

NOTES

This command is part of the xCAT software product.

SEE ALSO

genimage(1)|genimage.1, packimage(1)|packimage.1

rmkit.1

NAME

rmkit - Remove Kits from xCAT

SYNOPSIS

rmkit [**-?** | **-h** | **--help**] [**-v** | **--version**]

rmkit [**-V** | **--verbose**] [**-f** | **--force**] [**-t** | **--test**] *kitlist*

DESCRIPTION

The **rmkit** command removes kits on the xCAT management node from kit names.

Note: The xCAT support for Kits is only available for Linux operating systems.

OPTIONS

-h|--help

> Display usage message.

-V|--verbose

> Verbose mode.

-v|--version

> Command version.

-f|--force

> Remove this kit even there is any component in this kit is listed by osimage.kitcomponents. If this option is not specified, this kit will not be removed if any kit components listed in an osimage.kitcomponents

-t|--test

> Test if kitcomponents in this kit are used by osimage

kitlist

A comma delimited list of kits that are to be removed from the xCAT cluster. Each entry can be a kitname or kit basename. For kit basename, rmkit command will remove all the kits that have that kit basename.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To remove two kits from tarball files.

rmkit kit-test1,kit-test2

Output is similar to:

Kit kit-test1-1.0-Linux,kit-test2-1.0-Linux was successfully removed.

2. To remove two kits from tarball files even the kit components in them are still being used by osimages.

rmkit kit-test1,kit-test2 –force

Output is similar to:

Kit kit-test1-1.0-Linux,kit-test2-1.0-Linux was successfully removed.

3. To list kitcomponents in this kit used by osimage

rmkit kit-test1,kit-test2 -t

Output is similar to:

kit-test1-kitcomp-1.0-Linux is being used by osimage osimage-test Following kitcomponents are in use: kit-test1-kitcomp-1.0-Linux

SEE ALSO

lskit(1)|lskit.1, addkit(1)|addkit.1, addkitcomp(1)|addkitcomp.1, rmkitcomp(1)|rmkitcomp.1, chkkitcomp(1)|chkkitcomp.1

~

rmkitcomp.1

NAME

rmkitcomp - Remove Kit components from an xCAT osimage.

SYNOPSIS

rmkitcomp [**-? | -h | --help**] [**-v | --version**]

rmkitcomp [**-V | --verbose**] [**-u | --uninstall**] [**-f | --force**] [**--noscripts**] **-i** *osimage kitcompname_list*

DESCRIPTION

The **rmkitcomp** command removes kit components from an xCAT osimage. All the kit component attribute values that are contained in the osimage will be removed, and the kit comoponent meta rpm and package rpm could be uninstalled by <-u|-********-uninstall> option.

Note: The xCAT support for Kits is only available for Linux operating systems.

OPTIONS

-u|--uninstall

All the kit component meta rpms and package rpms in otherpkglist will be uninstalled during genimage for stateless image and updatenode for stateful nodes.

-h|--help

Display usage message.

-V|--verbose

Verbose mode.

-v|--version

Command version.

-f|--force

Remove this kit component from osimage no matter it is a dependency of other kit components.

--noscripts

Do not remove kitcomponent's postbootscripts from osimage

-i *osimage*

osimage name that include this kit component.

kitcompname_list

A comma-delimited list of valid full kit component names or kit component basenames that are to be removed from the osimage. If a basename is specified, all kitcomponents matching that basename will be removed from the osimage.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To remove a kit component from osimage

rmkitcomp -i rhels6.2-ppc64-netboot-compute comp-test1-1.0-1-rhels-6.2-ppc64

Output is similar to:

kitcomponents comp-test1-1.0-1-rhels-6.2-ppc64 were removed from osimage rhels6.2-ppc64-netboot-compute successfully

---

2. To remove a kit component even it is still used as a dependency of other kit component.

rmkitcomp -f -i rhels6.2-ppc64-netboot-compute comp-test1-1.0-1-rhels-6.2-ppc64

Output is similar to:

kitcomponents comp-test1-1.0-1-rhels-6.2-ppc64 were removed from osimage rhels6.2-ppc64-netboot-compute successfully

3. To remove a kit component from osimage and also remove the kit component meta RPM and package RPM. So in next genimage for statelss image and updatenode for statefull nodes, the kit component meta RPM and package RPM will be uninstalled.

rmkitcomp -u -i rhels6.2-ppc64-netboot-compute comp-test1-1.0-1-rhels-6.2-ppc64

Output is similar to:

kitcomponents comp-test1-1.0-1-rhels-6.2-ppc64 were removed from osimage rhels6.2-ppc64-netboot-compute successfully

SEE ALSO

lskit(1)|lskit.1, addkit(1)|addkit.1, rmkit(1)|rmkit.1, addkitcomp(1)|addkitcomp.1, chkkitcomp(1)|chkkitcomp.1

rmnimimage.1

NAME

rmnimimage - Use this xCAT command to remove NIM resources specified in an xCAT osimage definition.

SYNOPSIS

rmnimimage [-h|--help]

rmnimimage [-V|--verbose] [-f|--force] [-d|--delete] [-x|--xcatdef] [-M|--managementnode] [-s servicenoderange] osimage_name

DESCRIPTION

Use this xCAT command to remove the AIX resources specified in an xCAT osimage definition.

To list the contents of the xCAT osimage definition use the xCAT **lsdef** command ("lsdef -t osimage -l -o <osimage_name>"). **Before running the rmnimimage command you should be absolutely certain that you really want to remove the NIM resources specified in the xCAT osimage definition!**

The default behavior of this command is to remove all the NIM resources, except the lpp_source, on the xCAT management node in addition to the resources that were replicated on any xCAT service nodes.

This command may also be used to clean up individual xCAT service nodes and remove the xCAT osimage definitions.

The "nim -o remove" operation is used to remove the NIM resource definitions. If you wish to completely remove all the files and directories (left behind by the NIM command) you must specify the "-d" option when you run **rmnimimage**. The "-d" option will also remove the lpp_source resource.

If you wish to remove the NIM resource from one or more xCAT service nodes without removing the resources from the management node you can use the "-s <servicenoderange>" option. In this case the NIM resources specified in

the xCAT osimage definition will be removed from the service nodes ONLY. The NIM resources on the management node will not be removed.

If you wish to remove NIM resources on the management node only, you can specify the "-M" option.

If you wish to also remove the xCAT osimage definition you must specify the "-x" option.

This command will not remove NIM resources if they are currently being used in another xCAT osimage definition. To see which resources are common between osimages you can specify the "-V" option. You can override this check by specifying the "-f" option.

This command will not remove NIM resources if they are currently allocated. You must deallocate the resources before they can be removed. See the **xcat2nim** and **rmdsklsnode** commands for information on how to deallocate and remove NIM machine definitions for standalone and diskless nodes.

See the AIX NIM documentation for additional details on how to deallocate and remove unwanted NIM objects.

OPTIONS

-h |--help

> Display usage message.

-d|--delete

> Delete any files or directories that were left after the "nim -o remove" command was run. This option will also remove the lpp_source resouce and all files contained in the lpp_source directories. When this command completes all definitions and files will be completely erased so use with caution!

-f|--force

> Override the check for shared resources when removing an xCAT osimage.

-M|--managementnode

> Remove NIM resources from the xCAT management node only.

-s servicenoderange

> Remove the NIM resources on these xCAT service nodes only. Do not remove the NIM resources from the xCAT management node.

osimage_name

> The name of the xCAT osimage definition.

-V|--verbose

> Verbose mode. This option will display the underlying NIM commands that are being called.

-x|--xcatdef

> Remove the xCAT osimage definition.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

1. Remove all NIM resources specified in the xCAT "61image" definition.

rmnimimage 61image

The "nim -o remove" operation will be used to remove the NIM resource definitions on the management node as well as any service nodes where the resource has been replicated. This NIM operation does not completely remove all files and directories associated with the NIM resources.

2. Remove all the NIM resources specified by the xCAT "61rte" osimage definition. Delete ALL files and directories associated with the NIM resources. This will also remove the lpp_source resource.

rmnimimage -d 61rte

3. Remove all the NIM resources specified by the xCAT "614img" osimage definition and also remove the xCAT definition.

rmnimimage -x -d 614img

Note: When this command completes all definitions and files will be completely erased, so use with caution!

4. Remove the NIM resources specified in the "614dskls" osimage definition on the xcatsn1 and xcatsn2 service nodes. Delete all files or directories associated with the NIM resources.

rmnimimage -d -s xcatsn1,xcatsn2 614dskls

5. Remove the NIM resources specified in the "614old" osimage definition on the xCAT management node only.

rmnimimage -M -d 614old

FILES

/opt/xcat/bin/rmnimimage

NOTES

This command is part of the xCAT software product.

SEE ALSO

mknimimage(1)|mknimimage.1

rmvlan.1

NAME

rmvlan - It remves the vlan from the cluster.

SYNOPSIS

rmvlan *vlanid*

rmvlan [**-h** | **--help**]

---

rmvlan [**-v** | **--version**]

DESCRIPTION

The **rmvlan** command removes the given vlan ID from the cluster. It removes the vlan id from all the swithces involved, deconfigures the nodes so that vlan adaptor (tag) will be remved, cleans up /etc/hosts, DNS and database tables for the given vlan.

For added security, the root guard and bpdu guard were enabled for the ports in this vlan by mkvlan and chvlan commands. However, the guards will not be disabled by this command. To disable them, you need to use the switch command line interface. Please refer to the switch command line interface manual to see how to disable the root guard and bpdu guard for a port.

Parameters

vlanid is a unique vlan number.

OPTIONS

-h|--help Display usage message.

-v|--version The Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To remove vlan 3

   ```
   rmvlan 3
   ```

   If the nodes are KVM guest then the do the following after the vlan is removed: rpower node1,node2 off rmvm node1,node2

FILES

/opt/xcat/bin/rmvlan

SEE ALSO

mkvlan(1)|mkvlan.1, chvlan(1)|chvlan.1, lsvlan(1)|lsvlan.1

rmvm.1

NAME

rmvm - Removes HMC-, DFM-, IVM-, KVM-, Vmware- and zVM-managed partitions or virtual machines.

SYNOPSIS

rmvm [-h| –help]

rmvm [-v| –version]

rmvm [-V| –verbose] noderange [-r] [–service]

For KVM and Vmware:

rmvm [-p] [-f]

PPC (using Direct FSP Management) specific:

rmvm noderange

DESCRIPTION

The rmvm command removes the partitions specified in noderange. If noderange is an CEC, all the partitions associated with that CEC will be removed. Note that removed partitions are automatically removed from the xCAT database. For IVM-managed systems, care must be taken to not remove the VIOS partition, or all the associated partitions will be removed as well.

For DFM-managed (short For Direct FSP Management mode) normal power machines, only partitions can be removed. No options is needed.

OPTIONS

-h Display usage message.

-v Command Version.

-V Verbose output.

-r Retain the data object definitions of the nodes.

--service Remove the service partitions of the specified CECs.

-p Purge the existence of the VM from persistant storage. This will erase all storage related to the VM in addition to removing it from the active virtualization configuration.

-p|--part Remove the specified partiton on normal power machine.

-f Force remove the VM, even if the VM appears to be online. This will bring down a live VM if requested.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To remove the HMC-managed partition lpar3, enter:

rmvm lpar3

Output is similar to:

lpar3: Success

2. To remove all the HMC-managed partitions associated with CEC cec01, enter:

rmvm cec01

Output is similar to:

```
lpar1: Success
lpar2: Success
lpar3: Success
```

3. To remove the HMC-managed service partitions of the specified CEC cec01 and cec02, enter:

rmvm cec01,cec02 –service

Output is similar to:

```
cec01: Success
cec02: Success
```

4. To remove the HMC-managed partition lpar1, but retain its definition, enter:

rmvm lpar1 -r

Output is similar to:

lpar1: Success

5. To remove a zVM virtual machine:

rmvm gpok4

Output is similar to:

```
gpok4: Deleting virtual server LNX4... Done
```

6. To remove a DFM-managed partition on normal power machine:

rmvm lpar1

Output is similar to:

```
lpar1: Done
```

FILES

/opt/xcat/bin/rmvm

SEE ALSO

mkvm(1)|mkvm.1, lsvm(1)|lsvm.1, chvm(1)|chvm.1

rmzone.1

NAME

rmzone - Removes a zone from the cluster.

SYNOPSIS

rmzone <zonename> [**-g**] [**-f**]

rmzone [**-h** | **-v**]

DESCRIPTION

The **rmzone** command is designed to remove a previously defined zone from the cluster. It will remove the zone entry in the zone table. It will remove the zone from the zonename attributes on the nodes that were assigned to the zone. Optionally, it will remove the zonename group from the nodes that were assigned to the zone. It will also remove the root ssh keys that were created for that zone on the Management Node. The rmzone command is only supported on Linux (No AIX support). The nodes are not automatically updated with new root ssh keys by rmzone. You must run updatenode -k or xdsh -K to the nodes to update the root ssh keys. The nodes new ssh key will be assigned from the defaultzone in the zone table, or if no entries in the zone table, the keys will come from /root/.ssh. Note: if any zones in the zone table, there must be one and only one defaultzone. Otherwise, errors will occur.

OPTIONS

-h | --help

> Displays usage information.

-v | --version

> Displays command version and build date.

-f | --force

> Used to remove a zone that is defined as current default zone. This should only be done if you are removing all zones, or you will adding a new zone or changing an existing zone to be the default zone.

-g | --assigngroup

> Remove the assigned group named **zonename** from all nodes assigned to the zone being removed.

-V | --Verbose

> Verbose mode.

Examples

*

> To remove zone1 from the zone table and the zonename attribute on all it's assigned nodes , enter:
>
> **rmzone** *zone1*

*

> To remove zone2 from the zone table, the zone2 zonename attribute, and the zone2 group assigned to all nodes that were in zone2, enter:
>
> **rmzone** *zone2* -g

*

> To remove zone3 from the zone table, all the node zone attributes and override the fact it is the defaultzone, enter:
>
> **rmzone** *zone3* -g -f

Files

/opt/xcat/bin/rmzone/

Location of the rmzone command.

SEE ALSO

L <mkzone(1)|mkzone.1>,L <chzone(1)|chzone.1>,L <xdsh(1)|xdsh.1>, updatenode(1)|updatenode.1

rnetboot.1

NAME

rnetboot - Cause the range of nodes to boot to network.

SYNOPSIS

rnetboot [**-V** | **--verbose**] [**-s** *boot_device_order*] [**-F**] [**-f**] *noderange* [**-m** *table.column==expectedstatus* [**-m** *table.column=~expectedstatus*]] [**-t** *timeout*] [**-r** *retrycount*]

rnetboot [**-h** | **--help**] [**-v** | **--version**]

zVM specific:

rnetboot noderange [**ipl=** *address*]

DESCRIPTION

The rnetboot command will do what is necessary to make each type of node in the given noderange boot from the network. This is usually used to boot the nodes stateless or to network install system p nodes.

OPTIONS

-s

Set the boot device order. Accepted boot devices are hd and net.

-F

Force reboot the system no matter what state the node is. By default, rnetboot will not reboot the node if node is in 'boot' state.

-f

Force immediate shutdown of the partition.

-m

Use one or multiple -m flags to specify the node attributes and the expected status for the node installation monitoring and automatic retry mechanism. The operators ==, !=, =~ and !~ are valid. This flag must be used with -t flag.

Note: if the "val" fields includes spaces or any other characters that will be parsed by shell, the "attr<oper-ator>val" needs to be quoted. If the operator is "!~", the "attr<operator>val" needs to be quoted using single quote.

-r

specify the number of retries that the monitoring process will perform before declare the failure. The default value is 3. Setting the retrycount to 0 means only monitoring the os installation progress and will not re-initiate the installation if the node status has not been changed to the expected value after timeout. This flag must be used with -m flag.

-t

Specify the the timeout, in minutes, to wait for the expectedstatus specified by -m flag. This is a required flag if the -m flag is specified.

-V

Verbose output.

-h

Display usage message.

-v

Command Version.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

```
rnetboot 1,3

rnetboot 14-56,70-203

rnetboot 1,3,14-56,70-203

rnetboot all,-129-256
```

```
rnetboot all -s hd,net

rnetboot all ipl=00c
```

SEE ALSO

nodeset(8)|nodeset.8

rollupdate.1

NAME

rollupdate - performs cluster rolling update

SYNOPSIS

cat *stanza-file* | **rollupdate** [-V | --verbose] [-t| --test]

rollupdate [-? | -h | --help | -v | --version]

DESCRIPTION

The **rollupdate** command creates and submits scheduler reservation jobs that will notify xCAT to shutdown a group of nodes, run optional out-of-band commands from the xCAT management node, and reboot the nodes. Currently, only LoadLeveler is supported as a job scheduler with **rollupdate**.

Input to the **rollupdate** command is passed in as stanza data through STDIN. Information such as the sets of nodes that will be updated, the name of the job scheduler, a template for generating job command files, and other control data are required. See /opt/xcat/share/xcat/rollupdate/rollupdate.input.sample and /opt/xcat/share/xcat/rollupdate/rollupdate_all.input.sample for stanza keywords, usage, and examples.

The **rollupdate** command will use the input data to determine each set of nodes that will be managed together as an update group. For each update group, a job scheduler command file is created and a reservation request is submitted. When the group of nodes becomes available and the scheduler activates the reservation, the xcatd daemon on the management node will be notified to begin the update process for all the nodes in the update group. If specified, prescripts will be run, an operating system shutdown command will be sent to each node, out-of-band operations can be run on the management node, and the nodes are powered back on.

The **rollupdate** command assumes that, if the update is to include rebooting stateless or statelite nodes to a new operating system image, the image has been created and tested, and that all relevant xCAT commands have been run for the nodes such that the new image will be loaded when xCAT reboots the nodes.

OPTIONS

-v|--version

Command Version.

-V|--verbose

Display additional progress and error messages. Output is also logged in /var/log/xcat/rollupdate.log.

-t|--test

Run the rollupdate command in test mode only to verify the output files that are created. No scheduler reservation requests will be submitted.

-?|-h|--help

Display usage message.

RETURN VALUE

0

The command completed successfully.

1

An error has occurred.

EXAMPLES

1. To run a cluster rolling update based on the information you have created in the file /u/admin/rolling_updates/update_all.stanza enter:

```
cat /u/admin/rolling_updates/update_all.stanza | rollupdate
```

FILES

/opt/xcat/bin/rollupdate /opt/xcat/share/xcat/rollupdate/rollupdate.input.sample /opt/xcat/share/xcat/rollupdate/ll.tmpl /opt/xcat/share/xcat/rollupdate/rollupdate_all.input.sample /opt/xcat/share/xcat/rollupdate/llall.tmpl /var/log/xcat/rollupdate.log

SEE ALSO

rpower.1

NAME

rpower - remote power control of nodes

SYNOPSIS

rpower *noderange* [**--nodeps**] [**on** | **onstandby** | **off** | **suspend** | **stat** | **state** | **reset** | **boot**] [**-m** *table.column==expectedstatus* [**-m** *table.column=~expectedstatus*]] [**-t** *timeout*] [**-r** *retrycount*]

rpower [**-h** | **--help** | **-v** | **--version**]

BMC (using IPMI) specific:

rpower *noderange* [**on** | **off** | **softoff** | **reset** | **boot** | **stat** | **state** | **status** | **wake** | **suspend** [**-w** *timeout*] [**-o**] [**-r**]]

PPC (with IVM or HMC) specific:

rpower *noderange* [**--nodeps**] {**of**}

CEC (with HMC) specific:

rpower *noderange* [**on** | **off** | **reset** | **boot** | **onstandby**]

LPAR (with HMC) specific:

rpower *noderange* [**on** | **off** | **stat** | **state** | **reset** | **boot** | **of** | **sms** | **softoff**]

CEC (using Direct FSP Management) specific:

rpower *noderange* [**onstandby** | **stat** | **state**] [**-T tooltype**]
rpower *noderange* [**on** | **off** | **resetsp**]

Frame (using Direct FSP Management) specific:

rpower *noderange* [**rackstandby** | **exit_rackstandby** | **stat** | **state** | **resetsp**]

LPAR (using Direct FSP Management) specific:

rpower *noderange* [**on** | **off** | **stat** | **state** | **reset** | **boot** | **of** | **sms**]

Blade (using Direct FSP Management) specific:

rpower *noderange* [**on** | **onstandby** | **off** | **stat** | **state** | **sms**]

Blade specific:

rpower *noderange* [**cycle** | **softoff**]

zVM specific:

rpower *noderange* [**on** | **off** | **reset** | **stat** | **softoff**]

DESCRIPTION

rpower controls the power for a single or range of nodes, via the out-of-band path.

OPTIONS

on

> Turn power on.

onstandby

> Turn power on to standby state

-T

> The value could be **lpar** or **fnm**. The tooltype value **lpar** is for xCAT and **fnm** is for CNM. The default value is "**lpar**". For cold start in the large cluster, it will save a lot of time if the admins use "**rpower** *noderange* **onstandby -T fnm**" to power on all the CECs from the management node through the **fnm** connections.

rackstandby

> Places the rack in the rack standby state. It requires that all CECs and DE be powered off before it will run.

exit_rackstandby

> Exit Rack standby will be the default state that a rack goes into when power is initially applied to the rack. It simply moves the BPA from Rack standby to both bpa's in standby state.

resetsp

> Reboot the service processor. If there are primary and secondary FSPs/BPAs of one cec/frame, it will reboot them almost at the sametime.

softoff

> Attempt to request clean shutdown of OS (may not detect failures in completing command)

off

> Turn power off.

suspend

> Suspend the target nodes execution.
>
> The **suspend** action could be run together with **-w -o -r**.
>
> Refer to the following steps to enable the **suspend** function:
>
> 1. Add the 'acpid' and 'suspend'(the suspend package is not needed on RHEL) package to the .pkglist of your osimage so that the required package could be installed correctly to your target system.
>
> 2. Add two configuration files for the base function:

```
/etc/pm/config.d/suspend
    S2RAM_OPTS="--force --vbe_save --vbe_post --vbe_mode"

/etc/acpi/events/suspend_event
    event=button/sleep.*
    action=/usr/sbin/pm-suspend
```

> 3. Add the hook files for your specific applications which need specific action before or after the suspend action.
>
> Refer to the 'pm-utils' package for how to create the specific hook files.

wake

Wake up the target nodes which is in **suspend** state.

Don't try to run **wake** against the 'on' state node, it would cause the node gets to 'off' state.

For some of xCAT hardware such as NeXtScale, it may need to enable S3 before using **wake**. The following steps can be used to enable S3. Please reference pasu(1)|pasu.1 for "pasu" usage.

```
[root@xcatmn home]# echo "set Power.S3Enable Enable" > power-setting
[root@xcatmn home]# pasu -b power-setting node01
node01: Batch mode start.
node01: [set Power.S3Enable Enable]
node01: Power.S3Enable=Enable
node01:
node01: Beginning intermediate batch update.
node01: Waiting for command completion status.
node01: Command completed successfully.
node01: Completed intermediate batch update.
node01: Batch mode completed successfully.

[root@xcatmn home]# pasu node01 show all|grep -i s3
node01: IMM.Community_HostIPAddress3.1=
node01: IMM.Community_HostIPAddress3.2=
node01: IMM.Community_HostIPAddress3.3=
node01: IMM.DNS_IP_Address3=0.0.0.0
node01: IMM.IPv6DNS_IP_Address3=::
node01: Power.S3Enable=Enable
```

stat | state

> Print the current power state/status.

reset

> Send a hard reset.

boot

> If off, then power on. If on, then hard reset. This option is recommended over **cycle**.

cycle

> Power off, then on.

of

> Boot the node to open firmware console mode.

sms

> Boot the node to open firmware SMS menu mode.

-m *table.column==expectedstatus* **-m** *table.column=~expectedstatus*

> Use one or multiple **-m** flags to specify the node attributes and the expected status for the node installation monitoring and automatic retry mechanism. The operators ==, !=, =~ and !~ are valid. This flag must be used with -t flag.

> Note: if the "val" fields includes spaces or any other characters that will be parsed by shell, the "attr<operator>val" needs to be quoted. If the operator is "!~", the "attr<operator>val" needs to be quoted using single quote.

--nodeps

> Do not use dependency table (default is to use dependency table). Valid only with **on|off|boot|reset|cycle** for blade power method and **on|off|reset|softoff** for hmc/fsp power method.

-r *retrycount*

specify the number of retries that the monitoring process will perform before declare the failure. The default value is 3. Setting the retrycount to 0 means only monitoring the os installation progress and will not re-initiate the installation if the node status has not been changed to the expected value after timeout. This flag must be used with -m flag.

-t *timeout*

Specify the the timeout, in minutes, to wait for the expectedstatus specified by -m flag. This is a required flag if the -m flag is specified.

Power off, then on.

-w *timeout*

To set the *timeout* for the **suspend** action to wait for the success.

-o

To specify that the target node will be power down if **suspend** action failed.

-r

To specify that the target node will be reset if **suspend** action failed.

-h | --help

Prints out a brief usage message.

-v | --version

Display the version number.

EXAMPLES

*

```
rpower node4,node5 stat

node4: on
node5: off
```

*

```
rpower node5 on

node5: on
```

SEE ALSO

noderange(3)|noderange.3, rcons(1)|rcons.1, rinv(1)|rinv.1, rvitals(1)|rvitals.1, rscan(1)|rscan.1

rscan.1

NAME

rscan - Collects node information from one or more hardware control points.

SYNOPSIS

rscan [-h\|–help]

rscan [-v\|–version]

rscan [-V\|–verbose] noderange [-u][-w][-x\|-z]

DESCRIPTION

The rscan command lists hardware information for each node managed by the hardware control points specified in noderange.

For the management module of blade, if the blade server is a Flex system P node, the fsp belongs to the blade server also will be scanned.

Note: The first line of the output always contains information about the hardware control point. When using the rscan command to generate output for HMC or IVM hardware control points, it provides the FSPs and BPAs as part of the output. The only exception is the rscan -u flag which provides updates made hardware control point in the xCAT database.

OPTIONS

-h Display usage message.

-v Command Version.

-V Verbose output.

-u Updates and then prints out node definitions in the xCAT database for CEC/BPA. It updates the existing nodes that contain the same mtms and serial number for nodes managed by the specified hardware control point. This primarily works with CEC/FSP and frame/BPA nodes when the node name is not the same as the managed system name on hardware control point (HMC), This flag will update the BPA/FSP node name definitions to be listed as the managed system name in the xCAT database.

For the Flex system manager, both the blade server and fsp object of xCAT will be updated if the mpa and slot id are matched to the object which has been defined in the xCAT database.

Note: only the matched object will be updated.

-w Writes output to xCAT database.

-x XML format.

-z Stanza formated output.

RETURN VALUE

```
0 The command completed successfully.

1 An error has occurred.
```

EXAMPLES

1. To list all nodes managed by HMC hmc01 in tabular format, enter:

rscan hmc01

Output is similar to:

```
type    name                         id   type-model  serial-number  address

hmc     hmc01                             7310-C05    10F426A        hmc01
fsp     Server-9117-MMA-SN10F6F3D         9117-MMA    10F6F3D        3.3.3.197
lpar    lpar3                        4    9117-MMA    10F6F3D
lpar    lpar2                        3    9117-MMA    10F6F3D
lpar    lpar1                        2    9117-MMA    10F6F3D
lpar    p6vios                       1    9117-MMA    10F6F3D
```

2. To list all nodes managed by IVM ivm02 in XML format and write the output to the xCAT database, enter:

rscan ivm02 -x -w

Output is similar to:

```xml
<Node>
  <cons></cons>
  <profile></profile>
  <parent></parent>
  <serial>10B7D1G</serial>
  <model>9133-55A</model>
  <node>Server-9133-55A-10B7D1G</node>
  <mgt>ivm</mgt>
  <nodetype>fsp</nodetype>
  <hcp>ivm02</hcp>
  <groups>fsp,all</groups>
  <id>10</id>
</Node>

<Node>
  <cons>ivm</cons>
  <profile>lpar01</profile>
  <parent>Server-9133-55A-10B7D1G</parent>
  <serial></serial>
  <model></model>
  <node>lpar01</node>
  <mgt>ivm</mgt>
  <nodetype>lpar,osi</nodetype>
  <hcp>ivm02</hcp>
  <groups>lpar,all</groups>
  <id>1</id>
<Node>

</Node>
  <cons>ivm</cons>
  <profile>lpar02</profile>
  <parent>Server-9133-55A-10B7D1G</parent>
  <serial></serial>
  <model></model>
  <node>lpar02</node>
  <mgt>ivm</mgt>
  <nodetype>lpar,osi</nodetype>
```

```
  <hcp>ivm02</hcp>
  <groups>lpar,all</groups>
  <id>2</id>
</Node>
```

3. To list all nodes managed by HMC hmc02 in stanza format and write the output to the xCAT database, enter:

rscan hmc02 -z -w

Output is similar to:

```
Server-9458-100992001Y_B:
  objtype=node
  nodetype=bpa
  id=2
  model=9458-100
  serial=992001Y
  hcp=hmc02
  profile=
  parent=
  groups=bpa,all
  mgt=hmc
  cons=

Server-9119-590-SN02C5F9E:
  objtype=node
  type=fsp
  id=10
  model=9119-590
  serial=02C5F9E
  hcp=hmc02
  profile=
  parent=Server-9458-100992001Y_B
  groups=fsp,all
  mgt=hmc
  cons=

lpar01:
  objtype=node
  nodetype=lpar,osi
  id=1
  model=
  serial=
  hcp=hmc02
  profile=lpar01
  parent=Server-9119-590-SN02C5F9E
  groups=lpar,all
  mgt=hmc
  cons=hmc

lpar02:
  objtype=node
  nodetype=lpar,osi
  id=2
  model=
  serial=
  hcp=hmc02
  profile=lpar02
  parent=Server-9119-590-SN02C5F9E
```

```
groups=lpar,all
mgt=hmc
cons=hmc
```

4. To update definitions of nodes, which is managed by hmc03, enter:

rscan hmc03 -u

Output is similar to:

```
#Updated following nodes:
type    name                            id      type-model  serial-number  address
fsp     Server-9125-F2A-SN0262672-B     3       9125-F2A    0262672        192.168.200.
→243
```

5. To collects the node information from one or more hardware control points on zVM AND populate the database with details collected by rscan:

rscan gpok2 -W

Output is similar to:

```
gpok2:
  objtype=node
  arch=s390x
  os=sles10sp3
  hcp=gpok3.endicott.ibm.com
  userid=LINUX2
  nodetype=vm
  parent=POKDEV61
  groups=all
  mgt=zvm
```

6. To scan the Flex system cluster:

rscan cmm01

Output is similar to:

```
type    name              id      type-model  serial-number  mpa       address
cmm     AMM680520153      0       789392X     100048A        cmm01     cmm01
blade   SN#YL10JH184067   1       789542X     10F752A        cmm01     12.0.0.9
xblade  SN#YL10JH184068   2       789542X     10F652A        cmm01     12.0.0.10
blade   SN#YL10JH184079   3       789542X     10F697A        cmm01     12.0.0.11
```

7. To update the Flex system cluster:

rscan cmm01 -u

Output is similar to:

```
cmm     [AMM680520153]          Matched To =>[cmm01]
blade   [SN#YL10JH184067]       Matched To =>[cmm01node01]
blade   [SN#YL10JH184079]       Matched To =>[cmm01node03]
```

FILES

/opt/xcat/bin/rscan

SEE ALSO

lsslp(1)|lsslp.1

rsetboot.1

SYNOPSIS

rsetboot *noderange* {**hd** | **net** | **cd** | **default** | **stat**}

rsetboot [**-h** | **--help** | **-v** | **--version**]

DESCRIPTION

rsetboot sets the boot media that should be used on the next boot of the specified nodes. After the nodes are booted with the specified device (e.g. via rpower(1)|rpower.1), the nodes will return to using the default boot device specified in the BIOS. Currently this command is only supported for IPMI nodes.

OPTIONS

hd

> Boot from the hard disk.

net

> Boot over the network, using a PXE or BOOTP broadcast.

cd

> Boot from the CD or DVD drive.

def | default

> Boot using the default set in BIOS.

stat

> Display the current boot setting.

EXAMPLES

1. Set nodes 1 and 3 to boot from the network on the next boot:

```
rsetboot node1,node3 net
```

2. Display the next-boot value for nodes 14-56 and 70-203:

```
rsetboot node[14-56],node[70-203] stat
```

3. Restore the next-boot value for these nodes back to their default set in the BIOS:

```
rsetboot node1,node3,node[14-56],node[70-203] default
```

SEE ALSO

rbootseq(1)|rbootseq.1

rspconfig.1

NAME

rspconfig - Configures nodes' service processors

SYNOPSIS

rspconfig [**-h** | **--help** | **-v** | **--version**]

BMC/MPA specific:

rspconfig *noderange* {**alert** | **snmpdest** | **community**}

rspconfig *noderange* **alert**={**on** | **enable** | **off** | **disable**}

rspconfig *noderange* **snmpdest**=*snmpmanager-IP*

rspconfig *noderange* **community**={**public**|*string*}

BMC specific:

rspconfig *noderange* {**ip** | **netmask** | **gateway** | **backupgateway** | **garp**}

rspconfig *noderange* **garp**={*time*}

MPA specific:

rspconfig *noderange* {**sshcfg** | **snmpcfg** | **pd1** | **pd2** | **network** | **swnet** | **ntp** | **textid** | **frame**}

rspconfig *noderange* **USERID**={**newpasswd**} **updateBMC**={**y** | **n**}

rspconfig *noderange* **sshcfg**={**enable** | **disable**}

rspconfig *noderange* **snmpcfg**={**enable** | **disable**}

rspconfig *noderange* **solcfg**={**enable** | **disable**}

rspconfig *noderange* **pd1**={**nonred** | **redwoperf** | **redwperf**}

rspconfig *noderange* **pd2**={**nonred** | **redwoperf** | **redwperf**}

rspconfig *noderange* **network**={[**ip**],[**host**],[**gateway**],[**netmask**]|*}

rspconfig *noderange* **initnetwork**={[**ip**],[**host**],[**gateway**],[**netmask**]|*}

rspconfig *noderange* **textid**={*|**textid**}

rspconfig *singlenode* **frame**={**frame_number**}

rspconfig *noderange* **frame**={*}

rspconfig *noderange* **swnet={[ip],[gateway],[netmask]}**

rspconfig *noderange* **ntp={[ntpenable],[ntpserver],[frequency],[v3]}**

FSP/CEC specific:

rspconfig *noderange* **{autopower | iocap | decfg | memdecfg | procdecfg | time | date | spdump | sysdump | network}**

rspconfig *noderange* **autopower={enable | disable}**

rspconfig *noderange* **iocap={enable | disable}**

rspconfig *noderange* **time={hh:mm:ss}**

rspconfig *noderange* **date={mm:dd:yyyy}**

rspconfig *noderange* **decfg={enable|disable:policyname,...}**

rspconfig *noderange* **procdecfg={configure|deconfigure:processingunit:id,...}**

rspconfig *noderange* **memdecfg={configure|deconfigure:processingunit:unit|bank:id,...>}**

rspconfig *noderange* **network={nic,*}**

rspconfig *noderange* **network={nic,[IP,][hostname,][gateway,][netmask]}**

rspconfig *noderange* **network={nic,0.0.0.0}**

rspconfig *noderange* **HMC_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **admin_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **general_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* ***_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **{hostname}**

rspconfig *noderange* **hostname={*|name}**

rspconfig *noderange* **--resetnet**

Flex system Specific:

rspconfig *noderange* **sshcfg={enable | disable}**

rspconfig *noderange* **snmpcfg={enable | disable}**

rspconfig *noderange* **network={[ip],[host],[gateway],[netmask]|*}**

rspconfig *noderange* **solcfg={enable | disable}**

rspconfig *noderange* **textid={*|textid}**

rspconfig *noderange* **cec_off_policy={poweroff | stayon}**

BPA/Frame Specific:

rspconfig *noderange* **{network}**

rspconfig *noderange* **network={nic,*}**

rspconfig *noderange* **network={nic,[IP,][hostname,][gateway,][netmask]}**

rspconfig *noderange* **network={nic,0.0.0.0}**

rspconfig *noderange* **HMC_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **admin_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **general_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* ***_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **{hostname}**

rspconfig *noderange* **hostname={*|name}**

rspconfig *noderange* **--resetnet**

FSP/CEC (using Direct FSP Management) Specific:

rspconfig *noderange* **HMC_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **admin_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **general_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* ***_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **{sysname}**

rspconfig *noderange* **sysname={* | name}**

rspconfig *noderange* **{pending_power_on_side}**

rspconfig *noderange* **pending_power_on_side={temp | perm}**

rspconfig *noderange* **{cec_off_policy}**

rspconfig *noderange* **cec_off_policy={poweroff | stayon}**

rspconfig *noderange* **{BSR}**

rspconfig *noderange* **{huge_page}**

rspconfig *noderange* **huge_page={NUM}**

rspconfig *noderange* **{setup_failover}**

rspconfig *noderange* **setup_failover={enable | disable}**

rspconfig *noderange* **{force_failover}**

rspconfig *noderange* **--resetnet**

BPA/Frame (using Direct FSP Management) Specific:

rspconfig *noderange* **HMC_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **admin_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **general_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* ***_passwd={currentpasswd,newpasswd}**

rspconfig *noderange* **{frame}**

rspconfig *noderange* **frame={*|frame_number}**

rspconfig *noderange* {**sysname**}

rspconfig *noderange* **sysname={* | name}**

rspconfig *noderange* {**pending_power_on_side**}

rspconfig *noderange* **pending_power_on_side={temp | perm}**

rspconfig *noderange* **--resetnet**

HMC Specific:

rspconfig *noderange* {**sshcfg**}

rspconfig *noderange* **sshcfg={enable | disable}**

rspconfig *noderange* **--resetnet**

DESCRIPTION

rspconfig configures various settings in the nodes' service processors. If only a keyword is specified, without the **=**, it displays the current value.

For options **autopower | iocap | decfg | memdecfg | procdecfg | time | date | spdump | sysdump | network**, user need to use *chdef -t site enableASMI=yes* to enable ASMI first.

OPTIONS

alert={on|enable|off|disable**}**

> Turn on or off SNMP alerts.

autopower={enable|disable**}**

> Select the policy for auto power restart. If enabled, the system will boot automatically once power is restored after a power disturbance.

backupgateway

> Get the BMC backup gateway ip address.

community={public|string**}**

> Get or set the SNMP commmunity value. The default is *public*.

date={mm:dd:yyy**}**

> Enter the current date.

decfg={enable|disable*:policyname,...*}

> Enables or disables deconfiguration policies.

frame={framenumber|*}

> Set or get frame number. If no framenumber and * specified, framenumber for the nodes will be displayed and updated in xCAAT database. If framenumber is specified, it only supports single node and the framenumber will be set for that frame. If * is specified, it supports noderange and all the frame numbers for the noderange will be read from xCAT database and set to frames. Setting the frame number is a disruptive command which requires all CECs to be powered off prior to issuing the command.

cec_off_policy={poweroff | stayon}

Set or get cec off policy after lpars are powered off. If no cec_off_policy value specified, the cec_off_policy for the nodes will be displayed. the cec_off_policy has two values: **poweroff** and **stayon**. **poweroff** means Power off when last partition powers off. **stayon** means Stay running after last partition powers off. If cec_off_policy value is specified, the cec off policy will be set for that cec.

HMC_passwd={currentpasswd,newpasswd}

Change the password of the userid **HMC** for CEC/Frame. If the CEC/Frame is the factory default, the currentpasswd should NOT be specified; otherwise, the currentpasswd should be specified to the current password of the userid **HMC** for the CEC/Frame.

admin_passwd={currentpasswd,newpasswd}

Change the password of the userid **admin** for CEC/Frame from currentpasswd to newpasswd. If the CEC/Frame is the factory default, the currentpasswd should NOT be specified; otherwise, the currentpasswd should be specified to the current password of the userid **admin** for the CEC/Frame.

general_passwd={currentpasswd,newpasswd}

Change the password of the userid **general** for CEC/Frame from currentpasswd to newpasswd. If the CEC/Frame is the factory default, the currentpasswd should NOT be specified; otherwise, the currentpasswd should be specified to the current password of the userid **general** for the CEC/Frame.

**** *_passwd**={currentpasswd,newpasswd}**

Change the passwords of the userids **HMC**, **admin** and **general** for CEC/Frame from currentpasswd to newpasswd. If the CEC/Frame is the factory default, the currentpasswd should NOT be specified; otherwise, if the current passwords of the userids **HMC**, **admin** and **general** for CEC/Frame are the same one, the currentpasswd should be specified to the current password, and then the password will be changed to the newpasswd. If the CEC/Frame is NOT the factory default, and the current passwords of the userids **HMC**, **admin** and **general** for CEC/Frame are NOT the same one, this option could NOT be used, and we should change the password one by one.

frequency

The NTP update frequency (in minutes).

garp=*time*

Get or set Gratuitous ARP generation interval. The unit is number of 1/2 second.

gateway

The gateway ip address.

hostname

Display the CEC/BPA system names.

BSR

Get Barrier Synchronization Register (BSR) allocation for a CEC.

huge_page

Query huge page information or request NUM of huge pages for CEC. If no value specified, it means query huge page information for the specified CECs, if a CEC is specified, the specified huge_page value NUM will be used as the requested number of huge pages for the CEC, if CECs are specified, it means to request the same NUM huge pages for all the specified CECs.

setup_failover={*enable*|*disable*}

Enable or disable the service processor failover function of a CEC or display status of this function.

force_failover

>   Force a service processor failover from the primary service processor to the secondary service processor.

hostname={*|*name*}

>   Set CEC/BPA system names to the names in xCAT DB or the input name.

iocap={*enable*|*disable*}

>   Select the policy for I/O Adapter Enlarged Capacity. This option controls the size of PCI memory space allocated to each PCI slot.

ip

>   The ip address.

memdecfg={*configure*|*deconfigure*:*processingunit*:*unit*|*bank*:*id,...*}

>   Select whether each memory bank should be enabled or disabled. State changes take effect on the next platform boot.

netmask

>   The subnet mask.

network={[*ip*],[*host*],[*gateway*],[*netmask*]|*}

>   For MPA: get or set the MPA network parameters. If '*' is specified, all parameters are read from the xCAT database.

>   For FSP of Flex system P node: set the network parameters. If '*' is specified, all parameters are read from the xCAT database.

initnetwork={[*ip*],[*host*],[*gateway*],[*netmask*]|*}

>   For MPA only. Connecting to the IP of MPA from the hosts.otherinterfaces to set the MPA network parameters. If '*' is specified, all parameters are read from the xCAT database.

network={*nic*,{[*ip*],[*host*],[*gateway*],[*netmask*]}|*}

>   Not only for FSP/BPA but also for IMM. Get or set the FSP/BPA/IMM network parameters. If '*' is specified, all parameters are read from the xCAT database. If the value of *ip* is '0.0.0.0', this *nic* will be configured as a DHCP client. Otherwise this *nic* will be configured with a static IP.

>   Note that IPs of FSP/BPAs will be updated with this option, user needs to put the new IPs to /etc/hosts manually or with xCAT command makehosts. For more details, see the man page of makehosts.

nonred

>   Allows loss of redundancy.

ntp={[*ntpenable*],[*ntpserver*],[*frequency*],[*v3*]}

>   Get or set the MPA Network Time Protocol (NTP) parameters.

ntpenable

>   Enable or disable NTP (enable|disable).

ntpserver

>   Get or set NTP server IP address or name.

pd1={**nonred | redwoperf | redwperf**}

>   Power Domain 1 - determines how an MPA responds to a loss of redundant power.

pd2={**nonred | redwoperf | redwperf**}

Power Domain 2 - determines how an MPA responds to a loss of redundant power.

procdecfg={*configure|deconfigure:processingunit:id,...*}

Selects whether each processor should be enabled or disabled. State changes take effect on the next platform boot.

redwoperf

Prevents components from turning on that will cause loss of power redundancy.

redwperf

Power throttles components to maintain power redundancy and prevents components from turning on that will cause loss of power redundancy.

snmpcfg={*enable|disable*}

Enable or disable SNMP on MPA.

snmpdest={*snmpmanager-IP*}

Get or set where the SNMP alerts should be sent to.

solcfg={*enable|disable*}

Enable or disable the sol on MPA (or CMM) and blade servers belongs to it.

spdump

Performs a service processor dump.

sshcfg={*enable|disable*}

Enable or disable SSH on MPA.

swnet={[*ip*],[*gateway*],[*netmask*]}

Set the Switch network parameters.

sysdump

Performs a system dump.

sysname

Query or set sysname for CEC or Frame. If no value specified, means to query sysname of the specified nodes. If '*' specified, it means to set sysname for the specified nodes, and the sysname values would get from xCAT datebase. If a string is specified, it means to use the string as sysname value to set for the specified node.

pending_power_on_side={*temp|perm*}

List or set pending power on side for CEC or Frame. If no pending_power_on_side value specified, the pending power on side for the CECs or frames will be displayed. If specified, the pending_power_on_side value will be set to CEC's FSPs or Frame's BPAs. The value 'temp' means T-side or temporary side. The value 'perm' means P-side or permanent side.

time={*hh:mm:ss*}

Enter the current time in UTC (Coordinated Universal Time) format.

textid={*|*textid*}

Set the blade or MPA textid. When using '*', the textid used is the node name specified on the command-line. Note that when specifying an actual textid, only a single node can be specified in the noderange.

USERID={*newpasswd*} **updateBMC=**{*y|n*}

Change the password of the userid **USERID** for CMM in Flex system cluster. The option *updateBMC* can be used to specify whether updating the password of BMCs that connected to the speified CMM. The value is 'y' by default which means whenever updating the password of CMM, the password of BMCs will be also updated. Note that there will be several seconds needed before this command complete.

If value * is specified for USERID and the object node is *Flex System X node*, the password used to access the BMC of the System X node through IPMI will be updated as the same password of the userid **USERID** of the CMM in the same cluster.

--resetnet

Reset the network interfaces of the specified nodes.

v3

Enable or disable v3 authentication (enable|disable).

-h | --help

Prints out a brief usage message.

-v, --version

Display the version number.

EXAMPLES

*

To setup new ssh keys on the Management Module mm:

```
B<rspconfig> mm snmpcfg=enable sshcfg=enable
```

*

To turn on SNMP alerts for node5:

rspconfig *node5* **alert=on**

```
node5: Alerts: enabled
```

*

To display the destination setting for SNMP alerts for node4:

rspconfig *node4 snmpdest*

```
node4: BMC SNMP Destination 1: 9.114.47.227
```

*

To display the frame number for frame 9A00-10000001

rspconfig *9A00-10000001 frame*

```
9A00-10000001: 1
```

*

To set the frame number for frame 9A00-10000001

rspconfig *9A00-10000001 frame=2*

```
9A00-10000001: SUCCESS
```

*

To set the frame numbers for frame 9A00-10000001 and 9A00-10000002

rspconfig *9A00-10000001,9A00-10000002 frame=**

```
9A00-10000001: SUCCESS
9A00-10000002: SUCCESS
```

*

To display the MPA network parameters for mm01:

rspconfig *mm01 network*

```
mm01: MM IP: 192.168.1.47
mm01: MM Hostname: MM001125C31F28
mm01: Gateway: 192.168.1.254
mm01: Subnet Mask: 255.255.255.224
```

*

To change the MPA network parameters with the values in the xCAT database for mm01:

rspconfig *mm01 network=**

```
mm01: MM IP: 192.168.1.47
mm01: MM Hostname: mm01
mm01: Gateway: 192.168.1.254
mm01: Subnet Mask: 255.255.255.224
```

*

To change only the gateway parameter for the MPA network mm01:

rspconfig *mm01 network=,,192.168.1.1,*

```
mm01: Gateway: 192.168.1.1
```

*

To display the FSP network parameters for fsp01:

rspconfig *fsp01 network*

```
fsp01:
    eth0:
            IP Type: Dynamic
            IP Address: 192.168.1.215
            Hostname:
            Gateway:
            Netmask: 255.255.255.0

    eth1:
            IP Type: Dynamic
            IP Address: 192.168.200.51
            Hostname: fsp01
            Gateway:
            Netmask: 255.255.255.0
```

*

To change the FSP network parameters with the values in command line for eth0 on fsp01:

rspconfig *fsp01 network=eth0,192.168.1.200,fsp01,,255.255.255.0*

```
fsp01: Success to set IP address,hostname,netmask
```

*

To change the FSP network parameters with the values in the xCAT database for eth0 on fsp01:

rspconfig *fsp01 network=eth0,**

```
fsp01: Success to set IP address,hostname,gateway,netmask
```

*

To configure eth0 on fsp01 to get dynamic IP address from DHCP server:

rspconfig *fsp01 network=eth0,0.0.0.0*

```
fsp01: Success to set IP type to dynamic.
```

*

To get the current power redundancy mode for power domain 1 on mm01:

rspconfig *mm01 pd1*

```
mm01: Redundant without performance impact
```

*

To change the current power redundancy mode for power domain 1 on mm01 to non-redundant:

rspconfig *mm01 pd1=nonred*

```
mm01: nonred
```

*

To enable NTP with an NTP server address of 192.168.1.1, an update frequency of 90 minutes, and with v3 authentication enabled on mm01:

rspconfig *mm01 ntp=enable,192.168.1.1,90,enable*

```
mm01: NTP: disabled
mm01: NTP Server: 192.168.1.1
mm01: NTP: 90 (minutes)
mm01: NTP: enabled
```

*

To disable NTP v3 authentication only on mm01:

rspconfig *mm01 ntp=,,,disable*

```
mm01: NTP v3: disabled
```

*

To disable Predictive Failure and L2 Failure deconfiguration policies on mm01:

rspconfig *mm01 decfg=disable:predictive,L3*

```
mm01: Success
```

*

To deconfigure processors 4 and 5 of Processing Unit 0 on mm01:

rspconfig *mm01 procedecfg=deconfigure:0:4,5*

```
mm01: Success
```

*

To check if CEC sysname set correct on mm01:

rspconfig *mm01 sysname*

```
mm01: mm01
```

rspconfig *mm01 sysname=cec01*

```
mm01: Success
```

rspconfig *mm01 sysname*

```
mm01: cec01
```

*

To check and change the pending_power_on_side value of cec01's fsps:

rspconfig *cec01 pending_power_on_side*

```
cec01: Pending Power On Side Primary: temp
cec01: Pending Power On Side Secondary: temp
```

rspconfig *cec01 pending_power_on_side=perm*

```
cec01: Success
```

rspconfig *cec01 pending_power_on_side*

```
cec01: Pending Power On Side Primary: perm
cec01: Pending Power On Side Secondary: perm
```

*

To show the BSR allocation for cec01:

rspconfig *cec01 BSR*

```
cec01: Barrier Synchronization Register (BSR)
cec01: Number of BSR arrays: 256
cec01: Bytes per BSR array : 4096
cec01: Available BSR array : 0
cec01: Partition name: BSR arrays
cec01: lpar01        : 32
cec01: lpar02        : 32
```

```
cec01: lpar03       : 32
cec01: lpar04       : 32
cec01: lpar05       : 32
cec01: lpar06       : 32
cec01: lpar07       : 32
cec01: lpar08       : 32
```

*

To query the huge page information for CEC1, enter:

rspconfig *CEC1 huge_page*

```
CEC1: Huge Page Memory
CEC1: Available huge page memory(in pages):    0
CEC1: Configurable huge page memory(in pages): 12
CEC1: Page Size (in GB):                       16
CEC1: Maximum huge page memory(in pages):      24
CEC1: Requested huge page memory(in pages):    15
CEC1: Partition name: Huge pages
CEC1: lpar1         : 3
CEC1: lpar5         : 3
CEC1: lpar9         : 3
CEC1: lpar13        : 3
CEC1: lpar17        : 0
CEC1: lpar21        : 0
CEC1: lpar25        : 0
CEC1: lpar29        : 0
```

*

To request 10 huge pages for CEC1, enter:

rspconfig *CEC1 huge_page=10*

```
CEC1: Success
```

*

To disable service processor failover for cec01, in order to complete this command, the user should power off cec01 first:

rspconfig *cec01 setup_failover*

```
cec01: Failover status: Enabled
```

rpower *cec01 off*

rspconfig *cec01 setup_failover=disable*

```
cec01: Success
```

rspconfig *cec01 setup_failover*

```
cec01: Failover status: Disabled
```

*

To force service processor failover for cec01:

lshwconn *cec01*

```
cec01: 192.168.1.1: LINE DOWN
cec01: 192.168.2.1: sp=primary,ipadd=192.168.2.1,alt_ipadd=unavailable,
↪state=LINE UP
cec01: 192.168.1.2: sp=secondary,ipadd=192.168.1.2,alt_ipadd=unavailable,
↪state=LINE UP
cec01: 192.168.2.2: LINE DOWN
```

rspconfig *cec01 force_failover*

```
cec01: Success.
```

lshwconn *cec01*

```
cec01: 192.168.1.1: sp=secondary,ipadd=192.168.1.1,alt_ipadd=unavailable,
↪state=LINE UP
cec01: 192.168.2.1: LINE DOWN
cec01: 192.168.1.2: LINE DOWN
cec01: 192.168.2.2: sp=primary,ipadd=192.168.2.2,alt_ipadd=unavailable,
↪state=LINE UP
```

*

To deconfigure memory bank 9 and 10 of Processing Unit 0 on mm01:

rspconfig *mm01 memdecfg=deconfigure:bank:0:9,10*

```
mm01: Success
```

*

To reset the network interface of the specified nodes:

rspconfig -*****-resetnet*

Output is similar to:

```
Start to reset network..

Reset network failed nodes:

Reset network succeed nodes:
Server-8233-E8B-SN1000ECP-A,Server-9119-FHA-SN0275995-B,Server-9119-FHA-
↪SN0275995-A,

Reset network finished.
```

*

To update the existing admin password on fsp:

rspconfig *fsp admin_passwd=admin,abc123*

```
fsp: Success
```

*

To set the initial password for user HMC on fsp:

rspconfig *fsp HMC_passwd=,abc123*

```
fsp: Success
```

SEE ALSO

noderange(3)|noderange.3, rpower(1)|rpower.1, rcons(1)|rcons.1, rinv(1)|rinv.1, rvitals(1)|rvitals.1, rscan(1)|rscan.1, rflash(1)|rflash.1

rspreset.1

Name

rspreset - resets the service processors associated with the specified nodes

Synopsis

rspreset *noderange*

rspreset [**-h** | **--help** | **-v** | **--version**]

Description

rspreset resets the service processors associated with the specified nodes. It searches the **nodehm** table and associated tables to find the service processors associated with the nodes specified. If the node is a BMC-based node, the node's BMC will be reset. If the node is a blade, the blade's on board service processor will be reset.

Options

-h | **--help**

Print help.

-v | **--version**

Print version.

Examples

1. Reset the service processor that controls node5:

   ```
   rspreset node5
   ```

SEE ALSO

rpower(1)|rpower.1, nodehm(5)|nodehm.5

rvitals.1

Name

rvitals - remote hardware vitals

Synopsis

rvitals [**-h** | **--help** | **-v** | **--version**]

FSP/LPAR (with HMC) specific:

rvitals *noderange* {**temp** | **voltage** | **lcds** | **all**}

CEC/LPAR/Frame (using Direct FSP Management) specific:

rvitals *noderange* {**rackenv** | **lcds** | **all**} [**-V**| **--verbose**]

MPA specific:

rvitals *noderange* {**temp** | **voltage** | **wattage** | **fanspeed** | **power** | **leds** | **summary** | **all**}

Blade specific:

rvitals *noderange* {**temp** | **wattage** | **fanspeed** | **leds** | **summary** | **all**}

BMC specific:

rvitals *noderange* {**temp** | **voltage** | **wattage** | **fanspeed** | **power** | **leds** | **all**}

Description

rvitals retrieves hardware vital information from the on-board Service Processor for a single or range of nodes and groups.

Options

cputemp

   Retrieves CPU temperatures.

disktemp

   Retrieves HD back plane temperatures.

ambtemp

   Retrieves ambient temperatures.

---

temp

> Retrieves all temperatures.

voltage

> Retrieves power supply and VRM voltage readings.

fanspeed

> Retrieves fan speeds.

lcds

> Retrieves LCDs status.

rackenv

> Retrieves rack environmentals.

leds

> Retrieves LEDs status.

power

> Retrieves power status.

powertime

> Retrieves total power uptime. This value only increases, unless the Service Processor flash gets updated.

reboot

> Retrieves total number of reboots. This value only increases, unless the Service Processor flash gets updated.

state

> Retrieves the system state.

all

> All of the above.

-h | --help

> Print help.

-v | --version

> Print version.

Examples

rvitals *node5 all*

```
node5: CPU 1 Temperature: + 29.00 C (+ 84.2 F)
node5: CPU 2 Temperature: + 19.00 C (+ 66.2 F)
node5: DASD Sensor 1 Temperature: + 32.00 C (+ 89.6 F)
node5: System Ambient Temperature Temperature: + 26.00 C (+ 78.8 F)
node5: +5V Voltage: +  5.01V
node5: +3V Voltage: +  3.29V
node5: +12V Voltage: + 11.98V
node5: +2.5V Voltage: +  2.52V
node5: VRM1 Voltage: +  1.61V
```

```
node5: VRM2 Voltage: +  1.61V
node5: Fan 1 Percent of max:   100%
node5: Fan 2 Percent of max:   100%
node5: Fan 3 Percent of max:   100%
node5: Fan 4 Percent of max:   100%
node5: Fan 5 Percent of max:   100%
node5: Fan 6 Percent of max:   100%
node5: Current Power Status On
node5: Current LCD1: SuSE Linux
node5: Power On Seconds  11855915
node5: Number of Reboots   930
node5: System State Booting OS or in unsupported OS
```

SEE ALSO

rpower(1)|rpower.1, rinv(1)|rinv.1

sinv.1

NAME

sinv - Checks the software configuration of the nodes in the cluster.

SYNOPSIS

sinv [**-o** *output*] [**-p** *template path*] [**-t** *template count*] [**-s** *seed node*] [**-i**] [**-e**] [**-r**] [**-V**] [**--devicetype** *type_of_device*] [**-l** *userID*] [[**-f** *command file*] | [**-c** *command*]]

sinv [**-h** | **-v**]

DESCRIPTION

The **sinv** command is designed to check the configuration of the nodes in a cluster. The command takes as input command line flags, and one or more templates which will be compared against the output of the xdsh command, designated to be run by the -c or -f flag, on the nodes in the noderange.

The nodes will then be grouped according to the template they match and a report returned to the administrator in the output file designated by the -o flag, or to stdout.

sinv supports checking the output from the **rinv** or **xdsh** command.

The **sinv** command is an xCAT Distributed Shell Utility.

COMMAND SPECIFICATION:

The xdsh or rinv command to execute on the remote targets is specified by the -c flag, or by the **-f** flag

which is followed by the fully qualified path to a file containing the command.

Note: do not add | xdshcoll to the command on the command line or in the command file, it is automatically added by sinv.

The syntax for the **-c sinv** parameter is as follows:

"*command*[; *command*]..."

where *command* is the command to run on the remote target. Quotation marks are required to ensure that all commands in the list are executed remotely, and that any special characters are interpreted correctly on the remote target.

The **sinv** command does not work with any interactive commands, including those that read from standard input.

REMOTE SHELL COMMAND:

For xdsh, support is explicitly provided for AIX Remote Shell and OpenSSH, but any secure remote command that conforms to the IETF (Internet Engineering Task Force) Secure Remote Command Protocol can be used. See man **xdsh** for more details.

OPTIONS

-o | --output *report output file*

> Optional output file. This is the location of the file that will contain the report of the nodes that match, and do not match, the input templates. If the flag is not used, the output will go to stdout.

-p | --tp *template path*

> This is the path to the template file. The template contains the output of xdsh command, that has been run against a "seed" node, a node that contains the configuration that you would like all nodes in your noderange to match.

> The admin can create the template by running the xdsh command on the seed node, pipe to xdshcoll (required) and store the output in the template path. See examples.

> **Note:** The admin can also edit the template to remove any lines that they do not want checked.

> An alternative method is to use the [**-s** *seed node*] parameter, which will automatically build the template for you from the seed node named.

> If a template path file does not exist, and a seed node is not input, then sinv will automatically use the one node in the noderange as the seed node and build the template.

-t | --tc *template count*

> This count is the number of templates that the command will use to check for nodes matches. If the template in the template path does not match a node, the **sinv** will check additional templates up to the template count.

> For each node, it will compare the node against each template to see if there is a match. If there is no match, and we are not over the template count, then a new template will be created from the node output. This will result in having all nodes that match a given template reported in their group at the end of the run in the output file. If no template count is specified, 0 is the default, and all nodes will be compared against the first template.

-s | --seed *seed node*

> This is the node that will be used to build the first template that is stored in template path. You can use this parameter instead of running the command yourself to build the template.

> **Note:** If the template path file does not exists, and no seed node is supplied, the seed node automatically is one node in the noderange.

-i | --ignorefirst

> This flag suppresses the reporting of the nodes matching the first template. In very large systems, you may not want to show the nodes that have the correct configuration, since the list could contain thousands of nodes. This allows you to only report the nodes that do not match the required configuration.

-e | --exactmatch

This requires the check of node output against template to be an exact match. If this flag is not set, **sinv** checks to see if the return from the xdsh command to the nodes contain a match for each line in the input template (except for xdshcoll header and comments). If not in exactmatch mode, there can exist more lines in the xdsh return from the nodes.

For example, if running a "rpm -qa | grep xCAT" command, without exactmatch set, if the node containes more xCAT rpms that listed in the template, it would be considered a match, as long as all rpms listed in the template were on the node. With exactmatch set, the output must be identical to the template.

--devicetype *type_of_device*

Specify a user-defined device type that references the location of relevant device configuration file. The devicetype value must correspond to a valid device configuration file. xCAT ships some default configuration files for Ethernet switches and and IB switches under */opt/xcat/share/xcat/devicetype* directory. If you want to overwrite any of the configuration files, please copy it to */var/opt/xcat/* directory and cutomize it. For example, *base/IBSwitch/Qlogic/config* is the configuration file location if devicetype is specified as IBSwitch::Qlogic. xCAT will first search config file using */var/opt/xcat/* as the base. If not found, it will search for it using */opt/xcat/share/xcat/devicetype/* as the base.

-l | --user *user_ID*

Specifies a remote user name to use for remote command execution.

-c | --command

The xdsh or rinv command that will be run. The command should be enclosed in double quotes to insure correct shell interpretation. This parameter must only contain, the node range or the image path (Linux) or spot name for AIX. It cannot be used to set additional input flags to xdsh or rinv (for example -s,-T,-e). See examples below.

Note: do not add the | xdshcoll to the command, it is automatically added by sinv. sinv also automatically sets the -v flag for xdsh.

-f | --file

The file containing the xdsh or rinv command that will be run. This should be the fully qualified name of the file.

Note: do not add the | xdshcoll to the command in the file, it is automatically added by sinv.

-r | --remove

This flag indicates that generated templates should be removed at the at the end of the **sinv** command execution.

If the flag is input, then all templates that are generated by the **sinv**command, will be removed. If the first template is created by the admin, it will not be removed.

If the flag is not input, no templates will be removed. It is up to the admin to cleanup templates.

-h | --help

Displays usage information.

-v | --version

Displays xCAT release version.

-V | --Verbose

Verbose mode.

Examples

*

> To setup sinv.template (name optional) for input to the **sinv** command , enter:
>
> **xdsh** *node1,node2 "rpm -qa | grep ssh " | xdshcoll > /tmp/sinv.template*
>
> **Note: when setting up the template the output of xdsh must be piped** to xdshcoll, sinv processing depends on it.

*

> To setup rinv.template for input to the **sinv** command , enter:
>
> **rinv** *node1-node2 serial | xdshcoll > /tmp/rinv.template*
>
> **Note: when setting up the template the output of rinv must be piped** to xdshcoll, sinv processing depends on it.

*

> To execute **sinv** using the sinv.template generated above on the nodegroup, **testnodes** ,possibly generating up to two new templates, and removing all generated templates in the end, and writing output report to /tmp/sinv.output, enter:
>
> **sinv** * -c "xdsh testnodes rpm -qa | grep ssh" -p /tmp/sinv.template -t 2 -r -o /tmp/sinv.output*
>
> Note: do not add the pipe to xdshcoll on the -c flag, it is automatically added by the sinv routine.

*

> To execute **sinv** on noderange, node1-node4, using the seed node, node8, to generate the first template, using the xdsh command (-c), possibly generating up to two additional templates and not removing any templates at the end, enter:
>
> **sinv** *-c "xdsh node1-node4 lslpp -l | grep bos.adt" -s node8 -p /tmp/sinv.template -t 2 -o /tmp/sinv.output*

*

> To execute **sinv** on noderange, node1-node4, using the seed node, node8, to generate the first template, using the rinv command (-c), possibly generating up to two additional templates and removing any generated templates at the end, enter:
>
> **sinv** *-c "rinv node1-node4 serial" -s node8 -p /tmp/sinv.template -t 2 -r -o /tmp/rinv.output*

*

> To execute **sinv** on noderange, node1-node4, using node1 as the seed node, to generate the sinv.template from the xdsh command (-c), using the exact match option, generating no additional templates, enter:
>
> **sinv** *-c "xdsh node1-node4 lslpp -l | grep bos.adt" -s node1 -e -p /tmp/sinv.template -o /tmp/sinv.output*
>
> Note: the /tmp/sinv.template file must be empty, otherwise it will be used as an admin generated template.

*

> To execute **sinv** on the Linux osimage defined for cn1. First build a template from the /etc/hosts on the node. Then run sinv to compare. **xdsh** *cn1 "cat /etc/hosts" | xdshcoll * /tmp/sinv2/template"
>
> **sinv** *-c "xdsh -i /install/netboot/rhels6/ppc64/test_ramdisk_statelite/rootimg cat /etc/hosts" -e -t1 -p /tmp/sinv.template -o /tmp/sinv.output*

*

To execute **sinv** on the AIX NIM 611dskls spot and compare /etc/hosts to compute1 node, run the following:

xdsh *****compute1 "cat /etc/hosts" | xdshcoll ***** /tmp/sinv2/template"

sinv *-c "xdsh -i 611dskls cat /etc/hosts" -e -t1 -p /tmp/sinv.template -o /tmp/sinv.output*

*

To execute **sinv** on the device mswitch2 and compare to mswitch1

sinv *-c "xdsh mswitch enable;show version" -s mswitch1 -p /tmp/sinv/template -*******-devicetype IB-Switch::Mellanox -l admin -t 2*

Files

/opt/xcat/bin/sinv/

Location of the sinv command.

SEE ALSO

L <xdsh(1)|xdsh.1>, noderange(3)|noderange.3

snmove.1

NAME

snmove - Move xCAT compute nodes to a different xCAT service node.

SYNOPSIS

snmove *noderange* [**-V**] [**-l** | **--liteonly**] [**-d** | **--dest** *sn2*] [**-D** | **--destn** *sn2n*] [**-i** | **--ignorenodes**] [**-P** | **--postscripts** *script1,script2...|all*]

snmove [**-V**] [**-l** | **--liteonly**] **-s** | **--source** *sn1* [**-S** | **--sourcen** *sn1n*] [**-d** | **--dest** *sn2*] [**-D** | **--destn** *sn2n*] [**-i** | **--ignorenodes**] [**-P** | **--postscripts** *script1,script2...|all*]

snmove [**-h** | **--help** | **-v** | **--version**]

DESCRIPTION

The **snmove** command may be used to move a node or nodes from one service node to another backup service node.

The use of backup service nodes in an xCAT hierarchical cluster can help improve the overall reliability, availability, and serviceability of the cluster.

Before you run the **snmove** command it is assumed that the backup service node has been configured properly to manage the new node or nodes. (See the xCAT document named "Using xCAT Service Nodes with AIX" for information on how to set up backup AIX service nodes.).

The **snmove** command can use the information stored in the xCAT database or information passed in on the command line to determine the current service node and the backup service node.

To specify the primary and backup service nodes you can set the "servicenode" attribute of the node definitions.

The **servicenode** attribute is the hostname of the xCAT service node as it is known by the management node. The **xcatmaster** attribute is the hostname of the xCAT service node as known by the node. The **servicenode** attribute should be set to a comma-separated list so that the primary service node is first and the backup service node is second. The **xcatmaster** attribute must be set to the hostname of the primary service node as it is known by the node.

When the **snmove** command is run it modifies the xCAT database to switch the the primary server to the backup server.

It will also check the other services that are being used for the node (tftpserver, monserver, nfsserver, conserver), and if they were set to the original service node they will be changed to point to the backup service node.

By default the command will modify the nodes so that they will be able to be managed by the backup service node.

If the -i option is specified, the nodes themselves will not be modified.

You can also have postscripts executed on the nodes by using the -P option if needed.

The xCAT **snmove** command may also be used to synchronize statelite persistent files from the primary service node to the backup service node without actually moving the nodes to the backup servers.

If you run the command with the "-l" option it will attempt to use rsync to update the statelite persistent directory on the backup service node. This will only be done if the server specified in the "statelite" table is the primary service node.

When the **snmove** command is executed the new service node must be running but the original service node may be down.

Note: On a Linux cluster, for NFS statelite nodes that do not use external NFS server, if the original service node is down, the nodes it manages will be down too. You must run nodeset command and then reboot the nodes after running snmove. For stateless nodes and RAMDisk statelite nodes, the nodes will be up even if the original service node is down. However, make sure to run nodeset command if you decide to reboot the nodes later.

OPTIONS

-d|--dest

   Specifies the hostname of the new destination service node as known by (facing) the management node.

-D|--destn

   Specifies the hostname of the destination service node as known by (facing) the nodes.

-h|--help

   Display usage message.

-i|--ignorenodes

   No modifications will be made on the nodes. If not specified, several xCAT postscripts will be run on the nodes to complete the switch to the new service node.

-l|--liteonly

   Use this option to ONLY synchronize any AIX statelite files from the primary server to the backup server for the nodes. It will not do the actual moving of thre nodes the the backup servers.

-P|--postscripts

   Specifies a list of extra postscripts to be run on the nodes after the nodes are moved over to the new serive node. If 'all' is specified, all the postscripts defined in the postscripts table will be run for the nodes. The specified postscripts must be stored under /install/postscripts directory.

-s|--source

Specifies the hostname of the current (source) service node sa known by (facing) the management
node.

-S|--sourcen

Specifies the hostname of the current service node adapter as known by (facing) the nodes.

-V|--verbose

Verbose mode.

-v|--version

Command Version.

EXAMPLES

1. Move the nodes contained in group "group1" to the service node named "xcatsn02".

   snmove group1 -d xcatsn02 -D xcatsn02-eth1

2. Move all the nodes that use service node xcatsn01 to service node xcatsn02.

   snmove -s xcatsn01 -S xcatsn01-eth1 -d xcatsn02 -D xcatsn02-eth1

3. Move any nodes that have sn1 as their primary server to the backup service node set in the xCAT node definition.

   snmove -s sn1

4. Move all the nodes in the xCAT group named "nodegroup1" to their backup SNs.

   snmove nodegroup1

5. Move all the nodes in xCAT group "sngroup1" to the service node named "xcatsn2".

   snmove sngroup1 -d xcatsn2

6. Move all the nodes in xCAT group "sngroup1" to the SN named "xcatsn2" and run extra postscripts.

   snmove sngroup1 -d xcatsn2 -P test1

7. Move all the nodes in xCAT group "sngroup1" to the SN named "xcatsn2" and do not run anything on the nodes.

   snmove sngroup1 -d xcatsn2 -i

8. Synchronize any AIX statelite files from the primary server for compute03 to the backup server. This will not actually move the node to it's backup service node.

   snmove compute03 -l -V

FILES

/opt/xcat/sbin/snmove

SEE ALSO

noderange(3)|noderange.3

swapnodes.1

NAME

swapnodes - swap the location info in the db (all the attributes in the ppc table and the nodepos table) between 2 nodes. If swapping within a cec, it will assign the IO adapters that were assigned to the defective node to the available node.

SYNOPSIS

swapnodes [**-h**| **--help**]

swapnodes -c *current_node* **-f** *fip_node* [**-o**]

DESCRIPTION

This command is only for Power 775 using Direct FSP Management, and used in Power 775 Availability Plus.

The **swapnodes** command will keep the **current_node** name in the xCAT table, and use the *fip_node*'s hardware resource. Besides that, the IO adapters will be assigned to the new hardware resource if they are in the same CEC. So the swapnodes command will do 2 things:

```
(1)swap the location info in the db between 2 nodes:
 All the ppc table attributes (including hcp, id, parent, supernode and so on).
 All the nodepos table attributes(including rack, u, chassis, slot, room and so on).
(2)assign the I/O adapters from the defective node(the original current_node) to the
→available node(the original fip_node) if the nodes are in the same cec.
```

The **swapnodes** command shouldn't make the decision of which 2 nodes are swapped. It will just received the 2 node names as cmd line parameters.

After running **swapnodes** command, the order of the I/O devices may be changed after IO re-assignment, so the administrator needs to run **rbootseq** to set the boot string for the current_node. And then boot the node with the same image and same postscripts because they have the same attributes.

Without **-o** option, it's used to swap the location info in the db between 2 nodes. With **-o** option, it's used to move the *current_node* definition to *fip_node* (the 2nd octant), not move the *fip_node* definition to the 1st octant. If the two nodes are in a cec, it will assign the IO adapters that were assigned to the defective node to the available node. Originally, the *current_node* is a defective non-compute node, and *fip_node* is a avaible compute node. After the swapping, the *current_node* will be a available node.

OPTIONS

-h|--help

> Display usage message.

-c

> *current_node* – the defective non-compute node.

-f

> *fip_node* – a compute node which will be swapped as the non-compute node.

-o

one way. Only move the *current_node* definition to the *fip_node*'s hardware resource, and not move the fip_node definition to the *current_node*. And then the *current_node* will use the *fip_node*'s hardware resource, and the *fip_node* definition is not changed. if the two nodes are in the same CEC, the I/O adapter from the original *current_node* will be assigned to the *fip_node*.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1

To swap the service node attributes and IO assignments between sn1 and compute2 which are in the same cec, all the attributes in the ppc table and nodepos talbe of the two node will be swapped, and the the I/O adapters from the defective node (the original sn1) will be assigned to the available node (the original compute2). After the swapping, the sn1 will use the compute2's hardware resource and the I/O adapters from the original sn1.

```
swapnodes -c sn1 -f compute2
```

2

To swap the service node attributes and IO assignments between sn1 and compute2 which are NOT in the same cec, all the attributes in the ppc table and nodepos talbe of the two node will be swapped. After the swapping, the sn1 will use the compute2's hardware resource.

```
swapnodes -c sn1 -f compute2
```

3

Only to move the service node (sn1) definition to the compute node (compute2)'s hardware resource, and not move the compute2 definition to the sn1. After the swapping, the sn1 will use the compute2's hardware resource, and the compute2 definition is not changed.

```
swapnodes -c sn1 -f compute2 -o
```

FILES

$XCATROOT/bin/swapnodes

(The XCATROOT environment variable is set when xCAT is installed. The default value is "/opt/xcat".)

NOTES

This command is part of the xCAT software product.

SEE ALSO

lsvm(1)|lsvm.1, mkvm(1)|mkvm.1, chvm(1)|chvm.1

---

switchblade.1

SYNOPSIS

switchblade *MM* {**list** | **stat**}

switchblade *node* {**media** | **mt** | **kvm** | **video** | **both**} [*slot_num*]

switchblade [**-h** | **--help** | **-v** | **--version**]

DESCRIPTION

switchblade assigns the BladeCenter media tray and/or KVM to the specified blade, so that they can be used with that blade. If **list** or **stat** are specified instead, **switchblade** will display the current assignment. You can either specify a management module or a node (blade) to **switchblade**. If the latter, **switchblade** will determine the management module of the node.

OPTIONS

list | stat

> Display which blade the media tray and KVM are currently assigned to.

media | mt

> Assign the media tray to the specified blade.

kvm | video

> Assign the KVM (video display) to the specified blade.

both

> Assign both the media tray and the KVM to the specified blade.

slot_num

> The slot # of the blade that the resources should be assigned to. If not specified, it will use the slot # of the node specified.

EXAMPLES

1. Switch the media tray to be assigned to the blade in slot 4 (assume it is node4):

```
switchblade node4 media
```

Output will be like:

```
Media Tray slot: 4
```

SEE ALSO

rbootseq(1)|rbootseq.1

tabgrep.1

NAME

tabgrep - list table names in which an entry for the given node appears.

SYNOPSIS

tabgrep *nodename*

tabgrep [-*?* | -*h* | -***-help*]

DESCRIPTION

The tabgrep command displays the tables that contain a row for the specified node. Note that the row can either have that nodename as the key or it could have a group that contains the node as the key.

OPTIONS

-?|-h|--help

> Display usage message.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

*

> To display the tables that contain blade1:
>
> **tabgrep** *blade1*
>
> The output would be similar to:

```
nodelist
nodehm
mp
chain
hosts
mac
noderes
nodetype
```

FILES

/opt/xcat/bin/tabgrep

SEE ALSO

nodels(1)|nodels.1, tabdump(8)|tabdump.8

unregnotif.1

NAME

unregnotif - unregister a Perl module or a command that was watching for the changes of the desired xCAT database tables.

SYNOPSIS

unregnotif [-h| –help]

unregnotif [-v| –version]

*unregnotif *filename***

DESCRIPTION

This command is used to unregistered a Perl module or a command that was watching for the changes of the desired xCAT database tables.

Parameters

filename is the path name of the Perl module or command to be registered.

OPTIONS

-h | -help Display usage message.

-v | -version Command Version.

-V | -verbose Verbose output.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To unregistered a Perl module, enter:

```
unregnotif /opt/xcat/lib/perl/xCAT_monitoring/mycode.pm
```

2. To register a command, enter:

```
unregnotif /usr/bin/mycmd
```

FILES

/opt/xcat/bin/unregnotif

SEE ALSO

regnotif(1)|regnotif.1

updateSNimage.1

NAME

updateSNimage - Adds the needed Service Node configuration files to the install image.

SYNOPSIS

updateSNimage [-h | –help]

updateSNimage [-v | –version]

updateSNimage {-n} [-p]

DESCRIPTION

This command is used to add the Service Node configuration files to the install image. It will either copy them locally or scp them to a remote host.

OPTIONS

-h |--help Display usage message.

-v |--version Display xCAT version.

-n | --node A remote host name or ip address that contains the install image to be updated.

-p |--path Path to the install image.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To update the image on the local host.

updateSNimage -p /install/netboot/fedora8/x86_64/test/rootimg

2. To update the image on a remote host.

updateSNimage -n 9.112.45.6 -p /install/netboot/fedora8/x86_64/test/rootimg

updatenode.1

NAME

updatenode - Update nodes in an xCAT cluster environment.

SYNOPSIS

updatenode noderange [**-V** | **--verbose**] [**-F** | **--sync**] [**-f** | **--snsync**] [**-S** | **--sw**] [**-l** *userID*] [**-P** | **--scripts** [**script1,script2...**]] [**-s** | **--sn**] [**-A** | **--updateallsw**] [**-c** | **--cmdlineonly**] [**-d** alt_source_dir] [**--fanout**] [**-t** timeout} [**attr=val** [**attr=val...**]]

updatenode noderange [**-k** | **--security**] [**-t** timeout]

updatenode noderange [**-g** | **--genmypost**]

updatenode noderange [**-V** | **--verbose**] [**-t** timeout] [**script1,script2...**]

updatenode noderange [**-V** | **--verbose**] [**-f** | **--snsync**]

updatenode [**-h** | **--help**] [**-v** | **--version**]

DESCRIPTION

The updatenode command is run on the xCAT management node and can be used to perform the following node updates:

1

   Distribute and synchronize files.

2

   Install or update software on diskfull nodes.

3

   Run postscripts.

4

Update the ssh keys and host keys for the service nodes and compute nodes; Update the ca and credentials for the service nodes.

The default behavior when no options are input to updatenode will be to run the following options "-S", "-P" and "-F" options in this order. If you wish to limit updatenode to specific actions you can use combinations of the "-S", "-P", and "-F" flags.

For example, If you just want to synchronize configuration file you could specify the "-F" flag. If you want to synchronize files and update software you would specify the "-F" and "-S" flags. See the descriptions of these flags and examples below.

The flag "-k" (–security) can NOT be used together with "-S", "-P", and "-F" flags.

The flag "-f" (–snsync) can NOT be used together with "-S", "-P", and "-F" flags.

Note: In a large cluster environment the updating of nodes in an ad hoc manner can quickly get out of hand, leaving the system administrator with a very confusing environment to deal with. The updatenode command is designed to encourage users to handle cluster updates in a manner that is recorded and easily repeatable.

To distribute and synchronize files

The basic process for distributing and synchronizing nodes is:

*

Create a synclist file.

*

Indicate the location of the synclist file.

*

Run the updatenode command to update the nodes.

Files may be distributed and synchronized for both diskless and diskfull nodes. Syncing files to NFS-based statelite nodes is not supported.

More information on using the synchronization file function is in the following doc: Using_Updatenode.

Create the synclist file

The synclist file contains the configuration entries that specify where the files should be synced to. In the synclist file, each line is an entry which describes the location of the source files and the destination location for the files on the target node.

For more information on creating your synclist files and where to put them, read:

Sync-ing_Config_Files_to_Nodes

Run updatenode to synchronize the files

```
updatenode <noderange> -F
```

To install or update software

updatenode can be use to install or update software on the nodes. See the following documentation for setting up otherpkgs: Install_Additional_Packages

To install/update the packages, run:

```
updatenode <noderange> -S
```

For Linux systems:

It this is equivalent to running the following command:

```
updatenode noderange -P ospkgs,otherpkgs
```

It will update all the rpms specified in the .pkglist file and .otherpkgs.pkglist file. ospkgs postscript will normally remove all the existing rpm repositories before adding server:/install/<os>/<arch/ as the new repository. To preserve the existing repositories, you can run the following command instead:

updatenode noderange -P "ospkgs –keeprepo,otherpkgs"

For AIX systems:

Note: The updatenode command is used to update AIX diskfull nodes only. For updating diskless AIX nodes refer to the xCAT for AIX update documentation and use the xCAT mknimimage command. For information on updating software on AIX cluster: For diskful installs, read: XCAT_AIX_RTE_Diskfull_Nodes For diskless installs, read: XCAT_AIX_Diskless_Nodes

updatenode can also be used in Sysclone environment to push delta changes to target node. After capturing the delta changes from the golden client to management node, just run below command to push delta changes to target nodes.

```
updatenode <targetnoderange> -S
```

To run postscripts

The scripts must be copied to the /install/postscripts directory on the xCAT management node. (Make sure they are executable and world readable.)

To run scripts on a node you must either specify them on the command line or you must add them to the "postscripts" attribute for the node.

To set the postscripts attribute of the node (or group) definition you can use the xCAT chdef command. Set the value to be a comma separated list of the scripts that you want to be executed on the nodes. The order of the scripts in the list determines the order in which they will be run. You can use the lsdef command to check the postscript order.

Scripts can be run on both diskless and diskfull nodes.

To run all the customization scripts that have been designated for the nodes, (in the "postscripts and postbootscripts" attributes), type:

```
updatenode <noderange> -P
```

To run the "syslog" script for the nodes, type:

```
updatenode <noderange> -P syslog
```

To run a list of scripts, type:

```
updatenode <noderange> -P "script1 p1 p2,script2"
```

where p1 p2 are the parameters for script1.

The flag '-P' can be omitted when only scripts names are specified.

Note: script1,script2 may or may not be designated as scripts to automatically run on the node. However, if you want script1 and script2 to get invoked next time the nodes are deployed then make sure to add them to the "postscripts/postbootscripts" attribute in the database for the nodes.

Update security

The basic functions of update security for nodes:

*

Setup the ssh keys for the target nodes. It enables the management node and service nodes to ssh to the target nodes without password.

*

Redeliver the host keys to the target nodes.

*

Redeliver the ca and certificates files to the service node. These files are used to authenticate the ssl connection between xcatd's of management node and service node.

*

Remove the entries of target nodes from known_hosts file.

Set up the SSH keys

A password for the user who is running this command is needed to setup the ssh keys. This user must have the same uid and gid as the userid on the target node where the keys will be setup.

If the current user is root, roots public ssh keys will be put in the authorized_keys* files under roots .ssh directory on the node(s). If the current user is non-root, the user must be in the policy table and have credential to run the xdsh command. The non-root users public ssh keys and root's public ssh keys will be put in the authorized_keys* files under the non-root users .ssh directory on the node(s).

Handle the hierarchical scenario

When update security files for the node which is served by a service node, the service node will be updated automatically first, and then the target node.

The certificates files are needed for a service node to authenticate the ssl connections between the xCAT client and xcatd on the service node, and the xcatd's between service node and management node. The files in the directories /etc/xcat/cert/ and ~/.xcat/ will be updated.

Since the certificates have the validity time, the ntp service is recommended to be set up between management node and service node.

Simply running following command to update the security keys: updatenode *noderange* -k

PARAMETERS

noderange

A set of comma delimited xCAT node names and/or group names. See the xCAT "noderange" man page for details on additional supported formats.

script1,script2...

A comma-separated list of script names. The scripts must be executable and copied to the /install/postscripts directory. Each script can take zero or more parameters. If parameters are spcified, the whole list needs to be quoted by double quotes. For example:

"script1 p1 p2,script2"

[attr=val [attr=val...]]

Specifies one or more "attribute equals value" pairs, separated by spaces. Attr=val pairs must be specified last on the command line. The currently supported attributes are: "installp_bundle", "otherpkgs", "installp_flags", "emgr_flags" and "rpm_flags". These attribute are only valid for AIX software maintenance support.

OPTIONS

--fanout=*fanout_value*

Specifies a fanout value for the maximum number of concur- rently executing remote shell processes. Serial execution can be specified by indicating a fanout value of **1**. If **--fanout**is not specified, a default fanout value of **64** is used.

-A|--updateallsw

Install or update all software contained in the source directory. (AIX only)

-c|cmdlineonly

Specifies that the updatenode command should only use software maintenance information provided on the command line. This flag is only valid when using AIX software maintenance support.

-d alt_source_dir

Used to specify a source directory other than the standard lpp_source directory specified in the xCAT osimage definition. (AIX only)

-F|--sync

Specifies that file synchronization should be performed on the nodes. rsync and ssh must be installed and configured on the nodes. The function is not supported for NFS-based statelite installations. For NFS-based statelite installations to sync files, you should use the read-only option for files/directories listed in litefile table with source location specified in the litetree table.

-f|--snsync

Specifies that file synchronization should be performed to the service nodes that service the nodes in the noderange. This updates the service nodes with the data to sync to the nodes. rsync and ssh must be installed and configured on the service nodes. For hierarchy, this optionally can be done before syncing the files to the nodes with the -F flag. If the -f flag is not used, then the -F flag will sync the servicenodes before the nodes automatically. When installing nodes in a hierarchical cluster, this flag should be used to sync the service nodes before the install, since the files will be sync'd from the service node by the syncfiles postscript during the install. The function is not supported for NFS-based statelite installations. For statelite installations to sync files, you should use the read-only option for files/directories listed in litefile table with source location specified in the litetree table.

-g|--genmypost

Will generate a new mypostscript file for the nodes in the noderange, if site precreatemypostscripts is 1 or YES.

-h|--help

Display usage message.

-k|--security

Update the ssh keys and host keys for the service nodes and compute nodes; Update the ca and credentials to the service nodes. Never run this command to the Management Node, it will take down xcatd. You must be running updatenode as root to use the -k flag.

-l | --user *user_ID*

Specifies a non-root user name to use for remote command execution. This option is only available when running postscripts (-P) for AIX and Linux and updating software (-S) for Linux only. The non-root userid must be previously defined as an xCAT user. The userid sudo setup will have to be done by the admin on the node. This is not supported in a hiearchical cluster, that is the node is serviced by a service node. See the document Granting_Users_xCAT_privileges for required xcat/sudo setup.

-P|--scripts

Specifies that postscripts and postbootscripts should be run on the nodes. updatenode -P syncfiles is not supported. The syncfiles postscript can only be run during install. You should use updatenode <noderange> -F instead.

-S|--sw

Specifies that node software should be updated. In Sysclone environment, specifies pushing the delta changes to target nodes.

-s|--sn

Set the server information stored on the nodes in /opt/xcat/xcatinfo on Linux.

-t timeout

Specifies a timeout in seconds the command will wait for the remote targets to complete. If timeout is not specified it will wait indefinitely. updatenode -k is the exception that has a timeout of 10 seconds, unless overridden by this flag.

-v|--version

Command Version.

-V|--verbose

Verbose mode.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1

To perform all updatenode features for the Linux nodes in the group "compute":

updatenode compute

The command will: run any scripts listed in the nodes "postscripts and postbootscripts" attribute, install or update any software indicated in the /install/custom/install/<ostype>/profile.otherpkgs.pkglist (refer to the **To install or update software part**), synchronize any files indicated by the synclist files specified in the osimage "synclists" attribute.

2

To run postscripts,postbootscripts and file synchronization only on the node "clstrn01":

updatenode clstrn01 -F -P

3

Running updatenode -P with the syncfiles postscript is not supported. You should use updatenode -F instead.

Do not run:

updatenode clstrno1 -P syncfiles

Run:

updatenode clstrn01 -F

4

To run the postscripts and postbootscripts indicated in the postscripts and postbootscripts attributes on the node "clstrn01":

updatenode clstrn01 -P

5

To run the postscripts script1 and script2 on the node "clstrn01":

cp script1,script2 /install/postscripts

updatenode clstrn01 -P "script1 p1 p2,script2"

Since flag '-P' can be omitted when only script names are specified, the following command is equivalent:

updatenode clstrn01 "script1 p1 p2,script2"

p1 p2 are parameters for script1.

6

To synchronize the files on the node "clstrn01": Prepare the synclist file. For AIX, set the full path of synclist in the osimage table synclists attribute. For Linux, put the synclist file into the location: /install/custom/<inst_type>/<distro>/<profile>.<os>.<arch>.synclist Then:

updatenode clstrn01 -F

7

To perform the software update on the Linux node "clstrn01": Copy the extra rpm into the /install/post/otherpkgs/<os>/<arch>/* and add the rpm names into the /install/custom/install/<ostype>/profile.otherpkgs.pkglist . Then:

updatenode clstrn01 -S

8

To update the AIX node named "xcatn11" using the "installp_bundle" and/or "otherpkgs" attribute values stored in the xCAT database. Use the default installp, rpm and emgr flags.

updatenode xcatn11 -V -S

Note: The xCAT "xcatn11" node definition points to an xCAT osimage definition which contains the "installp_bundle" and "otherpkgs" attributes as well as the name of the NIM lpp_source resource.

9

To update the AIX node "xcatn11" by installing the "bos.cpr" fileset using the "-agQXY" installp flags. Also display the output of the installp command.

updatenode xcatn11 -V -S otherpkgs="I:bos.cpr" installp_flags="-agQXY"

Note: The 'I:' prefix is optional but recommended for installp packages.

10

To uninstall the "bos.cpr" fileset that was installed in the previous example.

updatenode xcatn11 -V -S otherpkgs="I:bos.cpr" installp_flags="-u"

11

To update the AIX nodes "xcatn11" and "xcatn12" with the "gpfs.base" fileset and the "rsync" rpm using the installp flags "-agQXY" and the rpm flags "-i –nodeps".

updatenode xcatn11,xcatn12 -V -S otherpkgs="I:gpfs.base,R:rsync-2.6.2-1.aix5.1.ppc.rpm" installp_flags="-agQXY" rpm_flags="-i --nodeps"

Note: Using the "-V" flag with multiple nodes may result in a large amount of output.

12

To uninstall the rsync rpm that was installed in the previous example.

updatenode xcatn11 -V -S otherpkgs="R:rsync-2.6.2-1" rpm_flags="-e"

13

Update the AIX node "node01" using the software specified in the NIM "sslbnd" and "sshbnd" installp_bundle resources and the "-agQXY" installp flags.

updatenode node01 -V -S installp_bundle="sslbnd,sshbnd" installp_flags="-agQXY"

14

To get a preview of what would happen if you tried to install the "rsct.base" fileset on AIX node "node42". (You must use the "-V" option to get the full output from the installp command.)

updatenode node42 -V -S otherpkgs="I:rsct.base" installp_flags="-apXY"

15

To check what rpm packages are installed on the AIX node "node09". (You must use the "-c" flag so updatenode does not get a list of packages from the database.)

updatenode node09 -V -c -S rpm_flags="-qa"

16

To install all software updates contained in the /images directory.

updatenode node27 -V -S -A -d /images

Note: Make sure the directory is exportable and that the permissions are set correctly for all the files. (Including the .toc file in the case of installp filesets.)

17

Install the interim fix package located in the /efixes directory.

updatenode node29 -V -S -d /efixes otherpkgs=E:IZ38930TL0.120304.epkg.Z

18

To uninstall the interim fix that was installed in the previous example.

updatenode xcatsn11 -V -S -c emgr_flags="-r -L IZ38930TL0"

19

To update the security keys for the node "node01"

updatenode node01 -k

20

To update the service nodes with the files to be synchronized to node group compute:

updatenode compute -f

21

To run updatenode with the non-root userid "user1" that has been setup as an xCAT userid with sudo on node1 to run as root, do the following: See Granting_Users_xCAT_privileges for required sudo setup.

updatenode node1 -l user1 -P syslog

22

In Sysclone environment, after capturing the delta changes from golden client to management node, to run updatenode to push these delta changes to target nodes.

updatenode target-node -S

FILES

/opt/xcat/bin/updatenode

wcons.1

Name

wcons - windowed remote console

Synopsis

wcons [**-t** | **--tile**=*n*] [*xterm-options*] *noderange*

wcons [**-h** | **--help** | **-v** | **--version**]

Description

wcons provides access to the remote node serial console of a single or range or nodes or groups.

wcons is a simple front-end to rcons in an xterm session for each console.

Options

-t | --tile=*n*

> Tile **wcons** windows from top left to bottom right. If *n* is spec- ified then tile *n* across. If *n* is not specified then tile to edge of screen. If tiled **wcons** windows reach bottom right, then the windows start at top left overlaying existing **wcons** windows.

-h | --help

> Print help.

-v | --version

> Print version.

xterm options

> See xterm(1). Any options other than those listed above are passed directly to xterm. **Note:** when given multiple nodes, wcons will override **-title** and tries to figure out optimal **-geometry**options for the xterms (however, **-geometry** can still be specified).

Files

nodehm table - xCAT node hardware management table. See nodehm(5)|nodehm.5 for further details. This is used to determine the console access method.

Examples

wcons *node1-node5*

wcons --tile --font=*nil2 all*

wcons -t *4 node1-node16*

wcons -f *vs* **-t** *4 node1-node4*

Bugs

Tile mode assumes that the width of the left window border is also the width of the right and bottom window border. Most window managers should not have a problem. If you really need support for a screwy window manager let me know.

See Also

noderange(3)|noderange.3, rcons(1)|rcons.1, xterm(1)

wkill.1

Name

wkill - kill windowed remote consoles

Synopsis

wkill [*noderange*]

wkill [**-h | --help | -v | --version**]

Description

wkill will kill the wcons windows on your $DISPLAY for a single or range or nodes or groups.

wkill was written because I'm too lazy to point and click off 64 windows.

wkill will only kill windows on your display and for only the noderange(3)|noderange.3 you specify. If no noderange(3)|noderange.3 is specified, then all wcons windows on your $DISPLAY will be killed.

Options

-h | --help

> Print help.

-v | --version

> Print version.

Examples

wkill *node1-node5*

See Also

noderange(3)|noderange.3, wcons(1)|wcons.1

wvid.1

Name

wvid - windowed remote video console for nodes

Synopsis

wvid *noderange*

Description

wvid provides access to the remote node video console of a single node, or range of nodes or groups. **wvid** provides a simple front-end to the hardware's remote console capability. Currently this command is supported for: blades, BMC/IMM, KVM, and Xen

---

The **nodehm.cons** attribute of the node determines the method used to open the console. See nodehm(5)|nodehm.5 for further details.

Options

No options are supported at this time.

Examples

1. To open video consoles for the 1st 2 nodes:

```
wvid node1,node2
```

See Also

noderange(3)|noderange.3, rcons(1)|rcons.1, wcons(1)|wcons.1

xCATWorld.1

NAME

xCATWorld - Sample client program for xCAT.

SYNOPSIS

xCATWorld {noderange}

DESCRIPTION

The xCATWorld program gives you a sample client program that interfaces to the /opt/xcat/lib/perl/xCAT_plugin/xCATWorld.pm plugin. For debugging purposes we have an Environment Variable XCATBYPASS. If export XCATBYPASS=yes, the client will call the plugin without going through the xcat daemon, xcatd.

OPTIONS

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1.To run , enter:

xCATWorld nodegrp1

FILES

/opt/xcat/bin/xCATWorld

NOTES

This command is part of the xCAT software product.

xcat2nim.1

NAME

xcat2nim - Use this command to create and manage AIX NIM definitions based on xCAT node, group and network object definitions.

SYNOPSIS

xcat2nim [-h|–help]

*xcat2nim [-V|–verbose] [-u|–update] [-l|–list] [-r|–remove] [-f|–force] [-t object-types] [-o object-names] [-a|–allobjects] [-p|–primarySN] [-b|–backupSN] [noderange] [attr=val [attr=val...]] *****

DESCRIPTION

The **xcat2nim** command uses xCAT node, group and network object definitions to create, update, list, or remove corresponding NIM definitions.

Before you create or update NIM definitions the xCAT definitions must be created and NIM must be configured.

The **xcat2nim** command uses xCAT database information, command line input, and default values to run the appropriate NIM commands.

The xCAT node, group and network definition names will correspond to the NIM machine, machine group and network definitions.

Note: The length of a NIM object name must be no longer than 39 characters.

To create or update a NIM definition you must provide the names of the xCAT definitions to use. The default behavior is to create new NIM definitions but not apply updates to existing definitions. If you wish to update existing NIM definitions then you must use the "update" option. If you wish to completely remove the old definition and re-create it you must use the "force" option.

The xCAT code uses the appropriate NIM commands to create the NIM definitions. To create definitions the "nim -o define" operation is used. To update definitions the "nim -o change" operation is used. If you wish to specify additional information to pass to the NIM commands you can use the "attr=val" support. The attribute names must correspond to the attributes supported by the relevant NIM commands. (For example. "netboot_kernel=mp")

If the object type you are creating is a node then the object names can be a noderange value.

If you are using xCAT service nodes the **xcat2nim** command will automatically determine the correct server for the node and create the NIM definitions on that server.

The **xcat2nim** command support for NIM networks is limited to creating and listing.

When creating network definitions the command will check to make sure the network definition (or it's equivalent) does not exist and then create the required NIM network, route and interface definitions. In some cases the equivalent network definition may exist using a different name. In this case a new definition WILL NOT be created.

To list the NIM definitions that were created you must specify the "list" option and the names of the xCAT objects that were used to create the NIM definitions. The **xcat2nim** command will list the corresponding NIM machine, machine group or network definitions using the "lsnim -l" command.

To remove NIM definitions you must specify the "remove" option and the names of the xCAT objects that were used to create the NIM definitions.

The remove("-r"), force("-f") and update("-u") options are not supported for NIM network definitions.

OPTIONS

-a|--all The list of objects will include all xCAT node, group and network objects.

attr=val [attr=val ...] Specifies one or more "attribute equals value" pairs, separated by spaces. Attr=val pairs must be specified last on the command line. The attribute names must correspond to the attributes supported by the relevant NIM commands. When providing attr=val pairs on the command line you must not specify more than one object type.

-b|--backupSN When using backup service nodes only update the backup. The default is to update both the primary and backup service nodes.

-f|--force The force option will remove the existing NIM definition and create a new one.

-h|--help Display the usage message.

-l|--list List NIM definitions corresponding to xCAT definitions.

-o object-names A set of comma delimited xCAT object names. Objects must be of type node, group, or network.

-p|--primarySN When using backup service nodes only update the primary. The default is to update both the primary and backup service nodes.

-r|--remove Remove NIM definitions corresponding to xCAT definitions.

-t object-types A set of comma delimited xCAT object types. Supported types include: node, group, and network.

Note: If the object type is "group", it means that the **xcat2nim** command will operate on a NIM machine group definition corresponding to the xCAT node group definition. Before creating a NIM machine group, all the NIM client nodes definition must have been created.

-u|--update Update existing NIM definitions based on xCAT definitions.

-V|--verbose Verbose mode.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To create a NIM machine definition corresponding to the xCAT node "clstrn01".

xcat2nim -t node -o clstrn01

2. To create NIM machine definitions for all xCAT node definitions.

xcat2nim -t node

3. Update all the NIM machine definitions for the nodes contained in the xCAT "compute" node group and specify attribute values that will be applied to each definition.

xcat2nim -u -t node -o compute netboot_kernel=mp cable_type="N/A"

4. To create a NIM machine group definition corresponding to the xCAT group "compute".

xcat2nim -t group -o compute

5. To create NIM network definitions corresponding to the xCAT "clstr_net" an "publc_net" network definitions. Also display verbose output.

xcat2nim -V -t network -o "clstr_net,publc_net"

6. To list the NIM definition for node clstrn02.

xcat2nim -l -t node clstrn02

7. To re-create a NIM machine definiton and display verbose output.

xcat2nim -V -t node -f clstrn05

8. To remove the NIM definition for the group "AIXnodes".

xcat2nim -t group -r -o AIXnodes

9. To list the NIM "clstr_net" definition.

xcat2nim -l -t network -o clstr_net

FILES

$XCATROOT/bin/xcat2nim

NOTES

This command is part of the xCAT software product.

SEE ALSO

mkdef(1)|mkdef.1

xcatchroot.1

NAME

xcatchroot - Use this xCAT command to modify an xCAT AIX diskless operating system image.

SYNOPSIS

xcatchroot -h ****

xcatchroot [-V] -i osimage_name cmd_string

DESCRIPTION

For AIX diskless images this command will modify the AIX SPOT resource using the chroot command. You must include the name of an xCAT osimage definition and the command that you wish to have run in the spot.

WARNING:

```
Be very careful when using this command!!!  Make sure you are
very clear about exactly what you are changing so that you do
not accidently corrupt the image.

As a precaution it is advisable to make a copy of the original
spot in case your changes wind up corrupting the image.
```

When you are done updating a NIM spot resource you should always run the NIM check operation on the spot.

```
nim -Fo check <spot_name>
```

The xcatchroot command will take care of any of the required setup so that the command you provide will be able to run in the spot chroot environment. It will also mount the lpp_source resource listed in the osimage definition so that you can access additional software that you may wish to install.

For example, assume that the location of the spot named in the xCAT osimage definition is /install/nim/spot/614spot/usr. The associated root directory in this spot would be /install/nim/spot/614spot/usr/lpp/bos/inst_root. The chroot is automatically done to this new root directory. The spot location is mounted on /.../inst_root/usr so that when your command is run in the chroot environment it is actually running commands from the spot usr location.

Also, the location of the lpp_source resource specified in the osimage definition will be mounted to a subdirectory of the spot /.../inst_root directory. For example, if the lpp_source location is /install/nim/lpp_source/614lpp_lpp_source then that would be mounted over /install/nim/spot/614spot/usr/lpp/bos/inst_root/lpp_source.

When you provide a command string to run make sure you give the full paths of all commands and files assuming the /.../inst_root directory is you root directory.

If you wish to install software from the lpp_source location you would provide a directory location of /lpp_source (or /lpp_source/installp/ppc or /lpp_source/RPMS/ppc etc.) See the example below.

Always run the NIM check operation after you are done updating your spot. (ex. "nim -o check <spot_name>")

OPTIONS

cmd_string

>   The command you wish to have run in the chroot environment. (Use a quoted string.)

-h |--help

>   Display usage message.

-i osimage_name

>   The name of the xCAT osimage definition.

-V |--verbose

>   Verbose mode.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

1) Set the root password to "cluster" in the spot so that when the diskless node boots it will have a root password set.

xcatchroot -i 614spot "/usr/bin/echo root:cluster | /usr/bin/chpasswd -c"

2. Install the bash rpm package.

xcatchroot -i 614spot "/usr/bin/rpm -Uvh /lpp_source/RPMS/ppc bash-3.2-1.aix5.2.ppc.rpm"

3. To enable system debug.

xcatchroot -i 614spot "bosdebug -D -M"

4. To set the "ipforwarding" system tunable.

xcatchroot -i 614spot "/usr/sbin/no -r -o ipforwarding=1"

FILES

/opt/xcat/bin/xcatchroot

NOTES

This command is part of the xCAT software product.

xcattest.1

NAME

xcattest - Run automated xCAT test cases.

SYNOPSIS

xcattest [**-?**|**-h**]

xcattest [**-f** *configure file*] [**-b** *case bundle list*] [**-l**]

xcattest [**-f** *configure file*] [**-t** *case list*]

xcattest [**-f** *configure file*] [**-c** *cmd list*] [**-l**]

DESCRIPTION

The xcattest command runs test cases to verify the xCAT functions, it can be used when you want to verify the xCAT functions for whatever reason, for example, to ensure the code changes you made do not break the existing commands; to run acceptance test for new build you got; to verify the xCAT snapshot build or development build before putting it onto your production system. The xcattest command is part of the xCAT package xCAT-test.

The root directory for the xCAT-test package is /opt/xcat/share/xcat/tools/autotest/. All test cases are in the sub directory *testcase*, indexed by the xCAT command, you can add your own test cases according to the test cases format below. The subdirectory *bundle* contains all the test cases bundles definition files, you can customize or create any test cases bundle file as required. The testing result information will be written into the subdirectory *result*, the timestamps are used as the postfixes for all the result files. xCAT-test package ships two configuration files template *aix.conf.template* and *linux.conf.template* for AIX and Linux environment, you can use the template files as the start point of making your own configuration file.

OPTIONS

-?|-h

> Display usage message.

-f *configure file*

> Specifies the configuration file, if not specified, the default configure file is /opt/xcat/share/xcat/tools/autotest/default.conf.

-b *case bundle list*

> Comma separated list of test cases bundle files, each test cases bundle can contain multiple lines and each line for one test case name.

-t *cases list*

> Comma separated list of test cases that will be run.

-c *cmd list*

> Comma separated list of commands which will be tested, i.e., all the test cases under the command sub directory will be run.

-l

> Display the test cases names specified by the flag -b, -t or -c.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

TEST CASE FORMAT

The xCAT-test test cases are in flat text format, the testing framework will parse the test cases line by line, here is an example of the test case:

```
#required, case name
start:case name
#optional, description of the test case
description: what the test case is for?
#optional, environment requirements
os:AIX/Linux
#optional, environment requirements
arch:ppc/x86
#optional, environment requirements
hcp:hmc/mm/bmc/fsp
#required, command need to run
cmd:comand
#optional, check return code of last executed command
check:rc == or != return code
#optional, check output of last executed command
check:output== or != or =~ or !~ output check string
end
```

Note: Each test case can have more than one *cmd* sections and each *cmd* section can have more than one *check:rc* sections and more than one *check:output* sections, the *output check string* can include regular expressions.

EXAMPLES

1. To run all test cases related command rpower:

   ```
   xcattest -c /tmp/config -c rpower
   ```

2. To run customized bundle:

   ```
   xcattest -l  > /tmp/custom.bundle
   Modify custom.bundle
   xcattest -b custom.bundle
   ```

3. To run specified test cases.

   ```
   xcattest -t lsdef_t_o_l_z
   ```

4. To add a new case to test chvm. In the example, we assume that the min_mem should not be equal to 16 in the lpar profile of computenode. The case name is chvm_custom. It create a test lpar named testnode firstly, that change the min_mem of the lpar to 16 using chvm, then check if min_mem have changed correctly. At last, the testnode be remove to ensure no garbage produced in the cases.

   ```
   add a new test case file in /opt/xcat/share/xcat/tools/autotest/chvm
   edit filename
   start:chvm_custom
   hcp:hmc
   cmd:lsvm $$CN > /tmp/autotest.profile
   check:rc==0
   cmd:mkdef -t node -o testnode mgt=hmc groups=all
   cmd:mkvm testnode -i $$MaxLparID -l $$CN
   check:rc==0
   cmd:perl -pi -e 's/min_mem=\d+/min_mem=16/g' /tmp/autotest.profile
   cmd:cat /tmp/autotest.profile|chvm testnode
   check:rc==0
   cmd:lsvm testnode
   check:output=~min_mem=16
   ```

```
cmd:rmvm testnode
cmd:rm -f /tmp/autotest.profile
end
```

INLINE FUNCTIONS

The xCAT-test testing framework provides some inline functions. The inline functions can be called in test cases as __FUNCTIONNAME(PARAMTERLIST)__ to get some necessary attributes defined in the configuration file. The inline functions can be used in *cmd* section and the *check:output* section.

1. **GETNODEATTR(nodename, attribute)** To get the value of specified node's attribute

2. **INC(digit)** To get value of digit+1.

For example, to run rscan command against the hardware control point of compute node specified in the configuration file:

```
  rscan __GETNODEATTR($$CN, hcp)__ -z
3. B<GETTABLEVALUE(keyname, key, colname, table)> To get the value of column where␣
→keyname == key in specified table.
```

FILES

/opt/xcat/bin/xcattest

xcoll.1

NAME

xcoll - Formats and consolidates the output of the **psh**, **rinv** commands.

SYNOPSIS

xcoll [**-n**] [**-c**]

DESCRIPTION

The **xcoll** command formats and consolidates output from the **psh,rinv** command. The **xcoll**command takes, as input, lines in the following format:

groupname: line of output from remote command, will use group name, if defined

The **xcoll** command formats the lines as follows and writes them to standard output. Assume that the output from node3 and node4 is identical:

```
===================================
node1 or nodegroup name
===================================
.
.
lines from psh for node1 with hostnames stripped off
```

```
.
.
==================================
node2 or nodegroup name
==================================
.
.
lines from psh for node2 with hostnames stripped off
.
.

==================================
node3, node4 or nodegroup name
==================================
.
.
lines from psh for node 3 with hostnames stripped off
.
.
```

OPTIONS

-c

> Display a total nodecount for each set of output.

-n

> Display output as nodenames instead of groupnames.

EXAMPLES

*

> To display the results of a command issued on several nodes, in the format used in the Description, enter:

> **psh** *node1,node2,node3 cat /etc/passwd* | **xcoll**

SEE ALSO

psh(1)|psh.1, xdshbak(1)|xdshbak.1 ,xdshcoll(1)|xdshcoll.1

xdcp.1

NAME

xdcp - Concurrently copies files to or from multiple nodes. In addition, provides an option to use rsync to update the files on the nodes, or to an installation image on the local node.

SYNOPSIS

xdcp *noderange* [[-**f** *fanout*] [-**L**] [-**l** *userID*] [-**o** *node_options*] [-**p**] [-**P**] [-**r** *node_remote_shell*] [-**R**] [-**t** *timeout*] [-**T**] [-**v**] [-**q**] [-**X** *env_list*] sourcefile.... targetpath

xdcp *noderange* [-**F** *rsync input file*]

xdcp *computenoderange* [-**s** -**F** *rsync input file*]

xdcp [-**i** *path to install image*] [-**F** *rsync input file*]

xdcp [-**h** | -**V** | -**q**]

DESCRIPTION

The **xdcp** command concurrently copies files to or from remote target nodes. The command issues a remote copy com- mand for each node or device specified. When files are pulled from a target, they are placed into the target_path with the name of the remote node or device appended to the copied source_file name. The /usr/bin/rcp command is the model for syntax and security. If using hierarchy, then xdcp runs on the service node that is servicing the compute node. The file will first be copied to the path defined in the site table, SNsyncfiledir attribute, or the default path /var/xcat/syncfiles on the service node, if the attribute is not defined. The -P flag will not automatically copy the files from the compute node to the Management node, hierarchically. There is a two step process, see -P flag. If the Management Node is target node, it must be defined in the xCAT database with nodetype=mn. When the **xdcp** command runs the Management Node as the target, it does not use remote commands but uses the local OS copy (cp) command.

REMOTE USER:

A user_ID can be specified for the remote copy command. Remote user specification is identical for the xdcp and xdsh commands. See the xdsh command for more information.

REMOTE COMMAND COPY: The **xdcp** command uses a configurable remote copy command to execute remote copies on remote targets. Support is explicitly provided for Remote Shell rcp command, the OpenSSH scp command and the /usr/bin/rsync command.

For node targets, the remote copy command is determined by the follow- ing order of precedence:

1. The **-r** flag.

2. The **/usr/bin/scp** command.

COMMAND EXECUTIONS The maximum number of concurrent remote copy command processes (the fanout) can be specified with the -f flag or the DSH_FANOUT environment variable. The fanout is only restricted by the number of remote shell commands that can be run in parallel. You can experiment with the DSH_FANOUT value on your management server to see if higher values are appropriate.

A timeout value for remote copy command execution can be specified with the -t flag or DSH_TIMEOUT environment variable. If any remote target does not respond within the timeout value, the xdcp command displays an error message and exits.

The -T flag provides diagnostic trace information for dcp command exe- cution. Default settings and the actual remote copy commands that are executed to the remote targets are displayed.

The xdcp command can be executed silently using the -Q flag; no target standard output or standard error is displayed.

OPTIONS

sourcefile...

Specifies the complete path for the file to be copied to or from the target. Multiple files can be specified. When used with the -R flag, only a single directory can be specified. When used with the -P flag, only a single file can be specified.

targetpath

If one source_file file, then it specifies the file to copy the source_file file to on the target. If multiple source_file files, it specifies the directory to copy the source_file files to on the target. If the -P flag is specified, the target_path is the local host location for the copied files. The remote file directory structure is recreated under target_path and the remote target name is appended to the copied source_file name in the target_path directory. Note: the targetpath directory must exist.

-f | --fanout *fanout_value*

Specifies a fanout value for the maximum number of concur- rently executing remote shell processes. Serial execution can be specified by indicating a fanout value of **1**. If **-f** is not specified, a default fanout value of **64** is used.

-F | --File *rsync input file*

Specifies the path to the file that will be used to build the rsync command. The format of the input file is as follows, each line contains:

```
<path to source file1> <path to source file2> ... ->  < path to destination␣
↪file/directory>
```

or

```
<path to source file> -> <path to destination file>
```

or

```
<path to source file> -> <path to destination directory ( must end in /)>
```

For example: /etc/password /etc/hosts -> /etc

```
/tmp/file2  ->  /tmp/file2

/tmp/file2  ->  /tmp/

/tmp/filex  ->  /tmp/source/filey

/etc/*  ->  /etc/
```

Running postscripts after files are sync'd to the nodes:

After you define the files to rsync, you can add an **EXECUTE:** clause in the synclist file. The **EXECUTE:** clause will list all the postscripts that you would like to run after the files are sync'd to the node. The postscript file must be of the form **filename.post**, where the <filename> is the <filename> is the from <filename>, reside in the same directory as **filename**, and be executable. If the file **filename** is rsync'd to the node, then the **filename.post** will automatically be run on the node. If the file **filename** is not updated on the node, the **filename.post** will not be run.

Putting the **filename.post** in the file list to rsync to the node is required for hierarchical clusters. It is optional for non-hierarchical clusters.

Another option is the **EXECUTEALWAYS:** clause in the synclist file. The **EXECUTEALWAYS:** will list all the postscripts that you would like to run after the files are sync'd to the nodes. These scripts will run whether or not any files are sync'd to the nodes. The scripts have no special format, but must contain the fully qualified path.

The scripts must be also added to the file list to rsync to the node for hierarchical clusters. It is optional for non-hierarchical clusters.

For example, your rsynclist file may look like this: /tmp/share/file2 -> /tmp/file2 /tmp/share/file2.post -> /tmp/file2.post /tmp/share/file3 -> /tmp/filex /tmp/share/file3.post -> /tmp/file3.post /tmp/myscript -> /tmp/myscript # the below are postscripts EXECUTE: /tmp/share/file2.post /tmp/share/file3.post EXECUTEALWAYS: /tmp/myscript

If /tmp/file2 and /tmp/file3 update /tmp/file2 and /tmp/filex on the node, then the postscripts /tmp/file2.post and /tmp/file3.post are automatically run on the node. /tmp/myscript will always be run on the node.

Another option is the **APPEND:** clause in the synclist file. The **APPEND:** clause is used to append the contents of the input file to an existing file on the node. The file to append **must** already exist on the node and not be part of the synclist that contains the **APPEND:** clause.

For example, your rsynclist file may look like this: /tmp/share/file2 -> /tmp/file2 /tmp/share/file2.post -> /tmp/file2.post /tmp/share/file3 -> /tmp/filex /tmp/share/file3.post -> /tmp/file3.post /tmp/myscript -> /tmp/myscript # the below are postscripts EXECUTE: /tmp/share/file2.post /tmp/share/file3.post EXECUTEALWAYS: /tmp/myscript APPEND: /etc/myappenddir/appendfile -> /etc/mysetup/setup /etc/myappenddir/appendfile2 -> /etc/mysetup/setup2

When you use the append script, the file (left) of the arrow is appended to the file right of the arrow. In this example, /etc/myappenddir/appendfile is appended to /etc/mysetup/setup file, which must already exist on the node. The /opt/xcat/share/xcat/scripts/xdcpappend.sh is used to accomplish this.

Another option is the **MERGE:** clause in the synclist file. The **MERGE:** clause is used to append the contents of the input file to /etc/passwd, /etc/group, or /etc/shadow on a Linux node. It is only supported for those files and only on Linux. You must not use both the APPEND and MERGE funcion for these three files. The processing could end up not creating the file you desire. The MERGE function is the preferred method, becuase APPEND only adds to the file. MERGE will add to the file but also insure there are no duplicate entries.

For example, your rsynclist file may look like this: /tmp/share/file2 -> /tmp/file2 /tmp/share/file2.post -> /tmp/file2.post /tmp/share/file3 -> /tmp/filex /tmp/share/file3.post -> /tmp/file3.post /tmp/myscript -> /tmp/myscript # the below are postscripts EXECUTE: /tmp/share/file2.post /tmp/share/file3.post EXECUTEALWAYS: /tmp/myscript APPEND:

/custom/mypasswd -> /etc/passwd /custom/mygroups -> /etc/group /custom/myshadow -> /etc/shadow

Note: no order can be assumed by the order that the EXECUTE,EXECUTEALWAYS and APPEND clause fall in the synclist file.

For more information on syncing files to node, read Sync-ing_Config_Files_to_Nodes

On Linux rsync always uses ssh remoteshell. On AIX, ssh or rsh is used depending on the site.useSSHonAIX attribute.

-h | --help

Displays usage information.

-i | --rootimg *install image*

Specifies the path to the install image on the local Linux node.

-o | --node-options *node_options*

Specifies options to pass to the remote shell command for node targets. The options must be specified within double quotation marks ("") to distinguish them from **xdsh** options.

-p | --preserve

Preserves the source file characteristics as implemented by the configured remote copy command.

-P | --pull

> Pulls (copies) the files from the targets and places them in the target_path directory on the local host. The target_path must be a directory. Files pulled from remote machines have ._target appended to the file name to distinguish between them. When the -P flag is used with the -R flag, ._target is appended to the directory. Only one file per invocation of the xdcp pull command can be pulled from the specified targets. Hierarchy is not automatically support yet. You must first pull the file to the Service Node and then pull the file to the Management node.

-q | --show-config

> Displays the current environment settings for all DSH Utilities commands. This includes the values of all environment variables and settings for all currently installed and valid contexts. Each setting is prefixed with *context*: to identify the source context of the setting.

-r | --node-rcp *node_remote_copy*

> Specifies the full path of the remote copy command used for remote command execution on node targets.

-R | --recursive *recursive*

> Recursively copies files from a local directory to the remote targets, or when specified with the -P flag, recursively pulls (copies) files from a remote directory to the local host. A single source directory can be specified using the source_file parameter.

-s *synch service nodes*

> Will only sync the files listed in the synclist (-F), to the service nodes for the input compute node list. The files will be placed in the directory defined by the site.SNsyncfiledir attribute, or the default /var/xcat/syncfiles directory.

-t | --timeout *timeout*

> Specifies the time, in seconds, to wait for output from any currently executing remote targets. If no output is available from any target in the specified *timeout*, **xdsh**displays an error and terminates execution for the remote targets that failed to respond. If *timeout* is not specified, **xdsh** waits indefinitely to continue processing output from all remote targets. When specified with the **-i** flag, the user is prompted for an additional timeout interval to wait for output.

-T | --trace

> Enables trace mode. The **xdcp** command prints diagnostic messages to standard output during execution to each target.

-v | --verify

> Verifies each target before executing any remote commands on the target. If a target is not responding, execution of remote commands for the target is canceled.

-V | --version

> Displays the **xdcp** command version information.

Environment Variables

DSH_ENVIRONMENT

> Specifies a file that contains environment variable definitions to export to the target before executing the remote command. This variable is overridden by the **-E** flag.

DSH_FANOUT

> Specifies the fanout value. This variable is overridden by the **-f** flag.

DSH_NODE_OPTS

> Specifies the options to use for the remote shell command with node targets only. This variable is overridden by the **-o** flag.

DSH_NODE_RCP

> Specifies the full path of the remote copy command to use to copy local scripts and local environment configuration files to node targets.

DSH_NODE_RSH

> Specifies the full path of the remote shell to use for remote command execution on node targets. This variable is overridden by the **-r** flag.

DSH_NODEGROUP_PATH

> Specifies a colon-separated list of directories that contain node group files for the **DSH** context. When the **-a** flag is specified in the **DSH** context, a list of unique node names is collected from all node group files in the path.

DSH_PATH

> Sets the command path to use on the targets. If **DSH_PATH** is not set, the default path defined in the profile of the remote *user_ID* is used.

DSH_SYNTAX

> Specifies the shell syntax to use on remote targets; **ksh** or **csh**. If not specified, the **ksh** syntax is assumed. This variable is overridden by the **-S** flag.

DSH_TIMEOUT

> Specifies the time, in seconds, to wait for output from each remote target. This variable is overridden by the **-t**flag.

Exit Status

Exit values for each remote copy command execution are displayed in messages from the xdcp command, if the remote copy command exit value is non-zero. A non-zero return code from a remote copy command indicates that an error was encountered during the remote copy. If a remote copy command encounters an error, execution of the remote copy on that tar- get is bypassed.

The xdcp command exit code is 0, if the xdcp command executed without errors and all remote copy commands finished with exit codes of 0. If internal xdcp errors occur or the remote copy commands do not complete successfully, the xdcp command exit value is greater than 0.

Security

The **xdcp** command has no security configuration requirements. All remote command security requirements - configuration, authentication, and authorization - are imposed by the underlying remote command configured for **xdsh**. The command assumes that authentication and authorization is configured between the local host and the remote targets. Interactive password prompting is not supported; an error is displayed and execution is bypassed for a remote target if password prompting occurs, or if either authorization or authentication to the remote target fails. Security configurations as they pertain to the remote environment and remote shell command are userdefined.

Examples

*

> To copy the /etc/hosts file from all nodes in the cluster to the /tmp/hosts.dir directory on the local host, enter:
>
> **xdcp** *all -P /etc/hosts /tmp/hosts.dir*
>
> A suffix specifying the name of the target is appended to each file name. The contents of the /tmp/hosts.dir directory are similar to:
>
> ```
> hosts._node1    hosts._node4    hosts._node7
> hosts._node2    hosts._node5    hosts._node8
> hosts._node3    hosts._node6
> ```

*

> To copy the directory /var/log/testlogdir from all targets in NodeGroup1 with a fanout of 12, and save each directory on the local host as /var/log._target, enter:
>
> **xdcp** *NodeGroup1 -f 12 -RP /var/log/testlogdir /var/log*

*

> To copy /localnode/smallfile and /tmp/bigfile to /tmp on node1 using rsync and input -t flag to rsync, enter:
>
> *xdcp node1 -r /usr/bin/rsync -o "-t" /localnode/smallfile /tmp/bigfile /tmp*

*

> To copy the /etc/hosts file from the local host to all the nodes in the cluster, enter:
>
> **xdcp** *all /etc/hosts /etc/hosts*

*

> To copy all the files in /tmp/testdir from the local host to all the nodes in the cluster, enter:
>
> **xdcp** *all /tmp/testdir/\ /tmp/testdir**

*

> To copy all the files in /tmp/testdir and it's subdirectories from the local host to node1 in the cluster, enter:
>
> **xdcp** *node1 -R /tmp/testdir /tmp/testdir*

*

> To copy the /etc/hosts file from node1 and node2 to the /tmp/hosts.dir directory on the local host, enter:
>
> **xdcp** *node1,node2 -P /etc/hosts /tmp/hosts.dir*

*

> To rsync the /etc/hosts file to your compute nodes:
>
> Create a rsync file /tmp/myrsync, with this line:
>
> /etc/hosts -> /etc/hosts
>
> or
>
> /etc/hosts -> /etc/ (last / is required)
>
> Run:
>
> **xdcp** *compute -F /tmp/myrsync*

---

*

To rsync all the files in /home/mikev to the compute nodes:

Create a rsync file /tmp/myrsync, with this line:

/home/mikev/* -> /home/mikev/ (last / is required)

Run:

xdcp *compute -F /tmp/myrsync*

*

To rsync to the compute nodes, using service nodes, the command will first rsync the files to the /var/xcat/syncfiles directory on the service nodes and then rsync the files from that directory to the compute nodes. The /var/xcat/syncfiles default directory on the service nodes, can be changed by putting a directory value in the site table SNsyncfiledir attribute.

Create a rsync file /tmp/myrsync, with this line:

/etc/hosts /etc/passwd -> /etc

or

/etc/hosts /etc/passwd -> /etc/

Run:

xdcp *compute -F /tmp/myrsync* to update the Compute Nodes

*

To rsync to the service nodes in preparation for rsyncing the compute nodes during an install from the service node.

Create a rsync file /tmp/myrsync, with this line:

/etc/hosts /etc/passwd -> /etc

Run:

xdcp *compute -s -F /tmp/myrsync* to sync the service node for compute

*

To rsync the /etc/file1 and file2 to your compute nodes and rename to filex and filey:

Create a rsync file /tmp/myrsync, with these line:

/etc/file1 -> /etc/filex

/etc/file2 -> /etc/filey

Run:

xdcp *compute -F /tmp/myrsync* to update the Compute Nodes

*

To rsync files in the Linux image at /install/netboot/fedora9/x86_64/compute/rootimg on the MN:

Create a rsync file /tmp/myrsync, with this line:

/etc/hosts /etc/passwd -> /etc

Run:

xdcp *-i /install/netboot/fedora9/x86_64/compute/rootimg -F /tmp/myrsync*

*

To define the Management Node in the database so you can use xdcp,run

xcatconfig -m

Files

SEE ALSO

xdsh(1)|xdsh.1, noderange(3)|noderange.3

xdsh.1

NAME

xdsh - Concurrently runs remote commands on multiple nodes (Management Node, Service Nodes, compute nodes), or an install image.

SYNOPSIS

xdsh *noderange* [**-B** *bypass*] [**--devicetype** *type_of_device*] [**-e**] [**-E** *environment_file*] [**-f** *fanout*] [**-L**] [**-l** *userID*] [**-m**] [**-o***node_options*] [**-Q**] [**-r** *node_remote_shell*] [**-s**] [**-S csh | ksh**] [**-t** *timeout*] [**-T**] [**-v**] [**-X** *env_list*] [**-z**] [**--sudo**] *command_list*

xdsh *noderange* [**-K**]

xdsh *noderange* [**-K**] [**-l** *userID*] **--devicetype** *type_of_device*

xdsh [**-i** *image path | nim image name*] *command_list*

xdsh *noderange* [**-c**]

xdsh [**-h** | **-V** | **-q**]

DESCRIPTION

The **xdsh** command runs commands in parallel on remote nodes and/or the Management Node. The **xdsh** command issues a remote shell command for each target specified, and returns the output from all targets, formatted so that command results from all nodes can be managed. If the command is to be executed on the Management Node, it does not use a remote shell command, but uses the local OS copy or shell command. The Management Node must be defined in the xCAT database. The best way to do this is to use the xcatconfig -m option. The **xdsh** command is an xCAT Distributed Shell Utility.

COMMAND SPECIFICATION:

The commands to execute on the targets are specified by the *command_list* **xdsh** parameter, or executing a local script using the **-e**flag.

The syntax for the *command_list* **xdsh** parameter is as follows:

command[; *command*]...

where *command* is the command to run on the remote target. Quotation marks are required to ensure that all commands in the list are executed remotely, and that any special characters are interpreted correctly on the remote target. A script

file on the local host can be executed on each of the remote targets by using the **-e**flag. If **-e** is specified, *command_list* is the script name and arguments to the script. For example:

xdsh hostname -e *script_filename* [*arguments*]...

The *script_filename* file is copied to a random filename in the **/tmp**directory on each remote target and then executed on the targets.

The **xdsh** command does not work with any interactive commands, including those that read from standard input.

REMOTE SHELL COMMAND:

The **xdsh** command uses a configurable remote shell command to execute remote commands on the remote targets. Support is explicitly provided for AIX Remote Shell and OpenSSH, but any secure remote command that conforms to the IETF (Internet Engineering Task Force) Secure Remote Command Protocol can be used.

The remote shell is determined as follows, in order of precedence:

1. The **-r** flag.

2. The **DSH_NODE_RSH** environment variable.

3. The default node remote shell as defined by the target *context*.

4. The **/usr/bin/ssh** command.

The remote shell options are determined as follows, in order of prece- dence:

1. The **-o** flag.

2. The **DSH_NODE_OPTS** environment variable.

REMOTE SHELL ENVIRONMENT:

The shell environment used on the remote target defaults to the shell defined for the *user_ID* on the remote target. The command syntax that **xdsh** uses to form the remote commands can be specified using the **-S**flag. If **-S** is not specified, the syntax defaults to **sh** syntax.

When commands are executed on the remote target, the path used is determined by the **DSH_PATH** environment variable defined in the shell of the current user. If **DSH_PATH** is not set, the path used is the remote shell default path. For example, to set the local path for the remote targets, use:

DSH_PATH=$PATH

The **-E** flag exports a local environment definition file to each remote target. Environment variables specified in this file are defined in the remote shell environment before the *command_list* is executed. The definition file should contain entries like the following

> **and be executable. One environment variable per line.** export NEWENVVARIABLE="yes" export
> ANOTHERENVVARIABLE="yes"

COMMAND EXECUTION:

The maximum number of concurrent remote shell command processes (the fanout) can be specified with the **-f** flag or with the **DSH_FANOUT**environment variable. The fanout is only restricted by the number of remote shell commands that can be run in parallel. You can experiment with the **DSH_FANOUT** value on your management server to see if higher values are appropriate.

A timeout value for remote command execution can be specified with the **-t** flag or with the **DSH_TIMEOUT** environment variable. If any remote target does not provide output to either standard output or standard error within the timeout value, **xdsh** displays an error message and exits.

If streaming mode is specified with the **-s** flag, output is returned as it becomes available from each target, instead of waiting for the *command_list* to complete on all targets before returning output. This can improve performance but causes the output to be unsorted.

The **-z** flag displays the exit code from the last command issued on the remote node in *command_list*. Note that OpenSSH behaves differently; it returns the exit status of the last remote command issued as its exit status. If the command issued on the remote node is run in the background, the exit status is not displayed.

The **-m** flag monitors execution of the **xdsh** command by printing status messages to standard output. Each status message is preceded by **dsh**.

The **-T** flag provides diagnostic trace information for the execution of the **xdsh** command. Default settings and the actual remote shell commands executed on the remote targets are displayed.

No error detection or recovery mechanism is provided for remote targets. The **xdsh** command output to standard error and standard output can be analyzed to determine the appropriate course of action.

COMMAND OUTPUT:

The **xdsh** command waits until complete output is available from each remote shell process and then displays that output before initiating new remote shell processes. This default behavior is overridden by the **-s** flag.

The **xdsh** command output consists of standard error and standard output from the remote commands. The **xdsh** standard output is the standard output from the remote shell command. The **xdsh** standard error is the standard error from the remote shell command. Each line is prefixed with the host name of the node that produced the output. The host name is followed by the **:** character and a command output line. A filter for displaying identical outputs grouped by node is provided separately. See the **xdshbak** command for more information.

A command can be run silently using the **-Q** flag; no output from each target's standard output or standard error is displayed.

SIGNALS:

Signal 2 (INT), Signal 3 (QUIT), and Signal 15 (TERM) are propagated to the commands executing on the remote targets.

Signal 19 (CONT), Signal 17 (STOP), and Signal 18 (TSTP) default to **xdsh**; the **xdsh** command responds normally to these signals, but the signals do not have an effect on remotely executing commands. Other signals are caught by **xdsh** and have their default effects on the **xdsh**command; all current child processes, through propagation to remotely running commands, are terminated (SIGTERM).

OPTIONS

-c | --cleanup

> This flag will have xdsh remove all files from the subdirectories of the the directory on the servicen- odes, where xdcp stages the copy to the compute nodes as defined in the site table SNsyncfiledir and nodesyncfiledir
>
> > attribute, when the target is a service node.
>
> It can also be used to remove the nodesyncfiledir directory on the compute nodes, which keeps the backup copies of files for the xdcp APPEND function support, if a compute node is the target.

-e | --execute

> Indicates that *command_list* specifies a local script filename and arguments to be executed on the remote targets. The script file is copied to the remote targets and then remotely executed with the given arguments. The **DSH_NODE_RCP** environment variables specify the remote copy command to use to copy the script file to node targets.

-E | --environment *environment_file*

> Specifies that the *environment_file* contains environment variable definitions to export to the target before executing the *command_list*.

--devicetype *type_of_device*

Specify a user-defined device type that references the location of relevant device configuration file. The devicetype value must correspond to a valid device configuration file. xCAT ships some default configuration files for Ethernet switches and and IB switches under */opt/xcat/share/xcat/devicetype* directory. If you want to overwrite any of the configuration files, please copy it to */var/opt/xcat/* directory and cutomize it. For example, *base/IBSwitch/Qlogic/config* is the configuration file location if devicetype is specified as IBSwitch::Qlogic. xCAT will first search config file using */var/opt/xcat/* as the base. If not found, it will search for it using */opt/xcat/share/xcat/devicetype/* as the base.

-f | --fanout *fanout_value*

Specifies a fanout value for the maximum number of concur- rently executing remote shell processes. Serial execution can be specified by indicating a fanout value of **1**. If **-f** is not specified, a default fanout value of **64** is used.

-h | --help

Displays usage information.

-i | --rootimg *install image*

For Linux, Specifies the path to the install image on the local node. For AIX, specifies the name of the osimage on the local node. Run lsnim for valid names. xdsh will chroot (xcatchroot for AIX) to this path and run the xdsh command against the install image. No other xdsh flags, environment variables apply with this input. A noderange is not accepted. Only runs on the local host, normally the Management Node. The command you run must not prompt for input, the prompt will not be returned to you, and it will appear that xdsh hangs.

-K | --ssh-setup

-K | --ssh-setup -l | --user *user_ID* **--devicetype** *type_of_device*

Set up the SSH keys for the user running the command to the specified node list. The userid must have the same uid, gid and password as the userid on the node

where the keys will be setup.

If the current user is root, roots public ssh keys will be put in the authorized_keys* files under roots .ssh directory on the node(s). If the current user is non-root, the user must be in the policy table and have credential to run the xdsh command. The non-root users public ssh keys and root's public ssh keys will be put in the authorized_keys* files under the non-root users .ssh directory on the node(s). Other device types, such as IB switch, are also supported. The device should be defined as a node and nodetype should be defined as switch before connecting. The xdsh -K command must be run from the Management Node.

-l | --user *user_ID*

Specifies a remote user name to use for remote command exe- cution.

-L | --no-locale

Specifies to not export the locale definitions of the local host to the remote targets. Local host locale definitions are exported by default to each remote target.

-m | --monitor

Monitors remote shell execution by displaying status messages during execution on each target.

-o | --node-options *node_options*

Specifies options to pass to the remote shell command for node targets. The options must be specified within double quotation marks ("") to distinguish them from **xdsh** options.

-q | --show-config

Displays the current environment settings for all DSH Utilities commands. This includes the values of all environment variables and settings for all currently installed and valid contexts. Each setting is prefixed with *context*: to identify the source context of the setting.

-Q | --silent

Specifies silent mode. No target output is written to stan- dard output or standard error. Monitoring messages are written to standard output.

-r | --node-rsh *node_remote_shell*

Specifies the path of the remote shell command used for remote command execution on node targets.

-s | --stream

Specifies that output is returned as it becomes available from each target, instead of waiting for the *command_list* to be completed on a target before returning output.

-S | --syntax csh | ksh

Specifies the shell syntax to be used on the remote target. If not specified, the **ksh** syntax is used.

--sudo | --sudo

Adding the –sudo flag to the xdsh command will have xdsh run sudo before running the command. This is particular useful when using the -e option. This is required when you input -l with a non-root user id and want that id to be able to run as root on the node. The non-root userid will must be previously defined as an xCAT user, see process for defining non-root ids in xCAT and setting up for using xdsh. The userid sudo setup will have to be done by the admin on the node. This includes, allowing all commands that you would like to run with xdsh by using visudo to edit the /etc/sudoers file. You must disabl ssh tty requirements by commenting out or removing this line in the /etc/sudoes file "#Defaults requiretty". See the document Granting_Users_xCAT_privileges for sudo setup requirements. This is not supported in a hierarical cluster, that is the nodes are serviced by servicenodes.

-t | --timeout *timeout*

Specifies the time, in seconds, to wait for output from any currently executing remote targets. If no output is available from any target in the specified *timeout*, **xdsh** displays an error and terminates execution for the remote targets that failed to respond. If *timeout* is not specified, **xdsh** waits indefinitely to continue processing output from all remote targets. The exception is the -K flag which defaults to 10 seconds.

-T | --trace

Enables trace mode. The **xdsh** command prints diagnostic messages to standard output during execution to each target.

-v | --verify

Verifies each target before executing any remote commands on the target. If a target is not responding, execution of remote commands for the target is canceled. When specified with the **-i** flag, the user is prompted to retry the verification request.

-V | --version

Displays the **xdsh** command version information.

-X *env_list*

Ignore **xdsh** environment variables. This option can take an argument which is a comma separated list of environment variable names that should **NOT** be ignored. If there is no argument to this option, or the argument is an empty string, all **xdsh** environment variables will be ignored. This option is useful when running **xdsh** from within other scripts when you don't want the user's environment affecting the behavior of xdsh.

-z | --exit-status

Displays the exit status for the last remotely executed non-asynchronous command on each target. If the command issued on the remote node is run in the background, the exit status is not displayed.

Exit values for each remote shell execution are displayed in messages from the **xdsh** command, if the remote shell exit values are non-zero. A non-zero return code from a remote shell indicates that an error was encountered in the remote shell. This return code is unrelated to the exit code of the remotely issued command. If a remote shell encounters an error, execution of the remote command on that target is bypassed.

The **xdsh** command exit code is **0** if the command executed without errors and all remote shell commands finished with exit codes of **0**. If internal **xdsh** errors occur or the remote shell commands do not complete successfully, the **xdsh** command exit value is greater than **0**. The exit value is increased by **1** for each successive instance of an unsuccessful remote command execution. If the remotely issued command is run in the background, the exit code of the remotely issued command is **0**.

Environment Variables

DEVICETYPE

Specify a user-defined device type. See **--devicetype** flag.

DSH_ENVIRONMENT

Specifies a file that contains environment variable definitions to export to the target before executing the remote command. This variable is overridden by the **-E** flag.

DSH_FANOUT

Specifies the fanout value. This variable is overridden by the **-f** flag.

DSH_NODE_OPTS

Specifies the options to use for the remote shell command with node targets only. This variable is overridden by the **-o** flag.

DSH_NODE_RCP

Specifies the full path of the remote copy command to use to copy local scripts and local environment configuration files to node targets.

DSH_NODE_RSH

Specifies the full path of the remote shell to use for remote command execution on node targets. This variable is overridden by the **-r** flag.

DSH_PATH

Sets the command path to use on the targets. If **DSH_PATH** is not set, the default path defined in the profile of the remote *user_ID* is used.

DSH_REMOTE_PASSWORD

If **DSH_REMOTE_PASSWORD** is set to the password of the userid (usually root) that will ssh to the node, then when you use the -K flag, you will not be prompted for a password.

DSH_SYNTAX

Specifies the shell syntax to use on remote targets; **ksh** or **csh**. If not specified, the **ksh** syntax is assumed. This variable is overridden by the **-S** flag.

DSH_TIMEOUT

Specifies the time, in seconds, to wait for output from each remote target. This variable is overridden by the **-t** flag.

Compatibility with AIX dsh

To provide backward compatibility for scripts written using dsh in AIX and CSM, a tool has been provide **group-files4dsh**, which will build node group files from the xCAT database that can be used by dsh. See man groupfiles4dsh.

Security

The **xdsh** command has no security configuration requirements. All remote command security requirements - configuration, authentication, and authorization - are imposed by the underlying remote command configured for **xdsh**. The command assumes that authentication and authorization is configured between the local host and the remote targets. Interactive password prompting is not supported; an error is displayed and execution is bypassed for a remote target if password prompting occurs, or if either authorization or authentication to the remote target fails. Security configurations as they pertain to the remote environment and remote shell command are userdefined.

Exit Status

The dsh command exit code is 0 if the command executed without errors and all remote shell commands finished with exit codes of 0. If internal dsh errors occur or the remote shell commands do not complete successfully, the dsh command exit value is greater than 0. The exit value is increased by 1 for each successive instance of an unsuccessful remote command execution. If the remotely issued command is run in the background, the exit code of the remotely issued command is 0.

Examples

*

To set up the SSH keys for root on node1, run as root:

xdsh *node1 -K*

*

To run the **ps -ef ** command on node targets **node1** and **node2**, enter:

xdsh *node1,node2 "ps -ef"*

*

To run the **ps** command on node targets **node1** and run the remote command with the -v and -t flag, enter:

xdsh *node1,node2 -o"-v -t" ps*=item *

To execute the commands contained in **myfile** in the **XCAT** context on several node targets, with a fanout of **1**, enter:

xdsh *node1,node2 -f 1 -e myfile*

*

To run the ps command on node1 and ignore all the dsh environment variable except the DSH_NODE_OPTS, enter:

xdsh *node1 -X\'DSH_NODE_OPTS' ps*

---

*

   To run on Linux, the xdsh command "rpm -qa | grep xCAT" on the service node fedora9 diskless image, enter:

   xdsh *-i /install/netboot/fedora9/x86_64/service/rootimg "rpm -qa | grep xCAT"*

*

   To run on AIX, the xdsh command "lslpp -l | grep bos" on the NIM 611dskls spot, enter:

   xdsh *-i 611dskls "/usr/bin/lslpp -l | grep bos"*

*

   To cleanup the servicenode directory that stages the copy of files to the nodes, enter:

   xdsh *****servicenoderange -c *****

*

   To define the QLogic IB switch as a node and to set up the SSH keys for IB switch **qswitch** with device configuration file **/var/opt/xcat/IBSwitch/Qlogic/config** and user name **username**, Enter

   chdef *-t node -o qswitch groups=all nodetype=switch*

   xdsh *qswitch -K -l username -*****-devicetype IBSwitch::Qlogic*

*

   To define the Management Node in the database so you can use xdsh, Enter

   xcatconfig -m

*

   To define the Mellanox switch as a node and run a command to show the ssh keys. **mswitch** with and user name **username**, Enter

   chdef *-t node -o mswitch groups=all nodetype=switch*

   xdsh *mswitch -l admin -*****-devicetype IBSwitch::Mellanox 'enable;configure terminal;show ssh server host-keys'*****

*

   To define a BNT Ethernet switch as a node and run a command to create a new vlan with vlan id 3 on the switch.

   chdef *myswitch groups=all*

   tabch *switch=myswitch switches.sshusername=admin switches.sshpassword=passw0rd switches.protocol=[ssh|telnet]*
         where *admin* and *passw0rd* are the SSH user name and password for the switch. If it is for Telnet, add *tn:* in front of the user name: *tn:admin*.

   <xdsh> *myswitch –devicetype EthSwitch::BNT 'enable;configure terminal;vlan 3;end;show vlan'*

*

   To run xdsh with the non-root userid "user1" that has been setup as an xCAT userid and with sudo on node1 and node2 to run as root, do the following, see xCAT doc on Granting_Users_xCAT_privileges:

   xdsh *node1,node2 -*****-sudo -l user1 "cat /etc/passwd"*****

Files

SEE ALSO

xdshbak(1)|xdshbak.1, noderange(3)|noderange.3, groupfiles4dsh(1)|groupfiles4dsh.1

xdshbak.1

NAME

xdshbak - Formats the output of the **xdsh** command.

SYNOPSIS

xdshbak [**-c** | **-x** | **-h** | **-q**]

DESCRIPTION

The **xdshbak** command formats output from the **xdsh** command. The **xdshbak** command takes, as input, lines in the following format:

```
host_name: line of output from remote command
```

The **xdshbak** command formats the lines as follows and writes them to standard output. Assume that the output from node3 and node4 is identical, and the **-c** (collapse) flag was specified:

```
HOSTS -------------------------------------------------------
node1
-------------------------------------------------------------
.
.
lines from xdsh for node1 with hostnames stripped off
.
.
HOSTS -------------------------------------------------------
node2
-------------------------------------------------------------
.
.
lines from xdsh for node2 with hostnames stripped off
.
.
HOSTS -------------------------------------------------------
node3, node4
-------------------------------------------------------------
.
.
lines from xdsh for node 3 with hostnames stripped off
.
.
```

When output is displayed from more than one node in collapsed form, the host names are displayed alphabetically. When output is not collapsed, output is displayed sorted alphabetically by host name.

If the **-q** quiet flag is not set then **xdshbak**command writes "." for each 1000 lines of output processed (to show progress), since it won't display the output until it has processed all of it.

If the **-x** flag is specified, the extra header lines that xdshbak normally displays for each node will be omitted, and the hostname at the beginning of each line is not stripped off, but **xdshbak** still sorts the output by hostname for easier viewing:

```
node1: lines from xdsh for node1
.
.
node2: lines from xdsh for node2
.
.
```

Standard Error

When the **xdshbak** filter is used and standard error messages are generated, all error messages on standard error appear before all standard output messages. This is true with and without the **-c** flag.

OPTIONS

-c

> If the output from multiple nodes is identical it will be collapsed and displayed only once.

-x

> Omit the extra header lines that xdshbak normally displays for each node. This provides more compact output, but xdshbak still sorts the output by node name for easier viewing. This option should not be used with **-c**.

-h

> Displays usage information.

-q

> Quiet mode, do not display "." for each 1000 lines of output.

EXAMPLES

*

> To display the results of a command issued on several nodes, in the format used in the Description, enter:

```
xdsh node1,node2,node3 cat /etc/passwd | xdshbak
```

*

> To display the results of a command issued on several nodes with identical output displayed only once, enter:

```
xdsh host1,host2,host3 pwd | xdshbak -c
```

*

> To display the results of a command issued on several nodes with compact output and be sorted alphabetically by host name, enter:

```
xdsh host1,host2,host3 date | xdshbak -x
```

SEE ALSO

xdsh(1)|xdsh.1, xcoll(1)|xcoll.1

xdshcoll.1

NAME

xdshcoll - Formats and consolidates the output of the **xdsh,sinv** commands.

SYNOPSIS

xdshcoll

DESCRIPTION

The **xdshcoll** command formats and consolidates output from the **xdsh,sinv** command. The **xdshcoll**command takes, as input, lines in the following format:

host_name: line of output from remote command

The **xdshcoll** command formats the lines as follows and writes them to standard output. Assume that the output from node3 and node4 is identical:

```
===================================
node1
===================================
.
.
lines from xdsh for node1 with hostnames stripped off
.
.


===================================
node2
===================================
.
.
lines from xdsh for node2 with hostnames stripped off
.
.


===================================
node3, node4
===================================
.
```

```
.
lines from xdsh for node 3 with hostnames stripped off
.
.
```

EXAMPLES

*

> To display the results of a command issued on several nodes, in the format used in the Description, enter:
>
> **xdsh** *node1,node2,node3 cat /etc/passwd* | **xdshcoll**

SEE ALSO

xdshbak(1)|xdshbak.1

xpbsnodes.1

NAME

xpbsnodes - PBS pbsnodes front-end for a noderange.

SYNOPSIS

xpbsnodes [{**noderange**}] [{**offline** | **clear** | **stat** | **state**}]

xpbsnodes [**-h** | **--help**] [**-v** | **--version**]

OPTIONS

-h Display usage message.

-v Command Version.

offline|off Take nodes offline.

clear|online|on Take nodes online.

stat|state Display PBS node state.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To display status of all PBS nodes, enter:

```
xpbsnodes all stat
```

FILES

/opt/torque/x86_64/bin/xpbsnodes

SEE ALSO

noderange(3)|noderange.3

man3

noderange.3

Name

noderange - syntax for compactly expressing a list of node names

Synopsis

Examples:

```
node1,node2,node8,node20,group1

node14-node56,node70-node203,group1-group10

node1,node2,node8,node20,node14-node56,node70-node203

node[14-56]

f[1-3]n[1-20]

all,-node129-node256,-frame01-frame03

/node.*

^/tmp/nodes

node10+5

10-15,-13

group1@group2

table.attribute<operator>value
```

Description

noderange is a syntax that can be used in most xCAT commands to conveniently specify a list of nodes. The result is that the command will be applied to a range of nodes, often in parallel.

noderange is a comma-separated list. Each token (text between commas) in the list can be any of the forms listed below:

Individual node or group:

```
node01
group1
```

A range of nodes or groups:

```
node01-node10   (equivalent to: node01,node02,node03,...node10)
node[01-10]     (same as above)
node01:node10   (same as above)
node[01:10]     (same as above)
f[1-2]n[1-3]    (equivalent to: f1n1,f1n2,f1n3,f2n1,f2n2,f2n3)
group1-group3   (equivalent to: group1,group2,group3)
(all the permutations supported above for nodes are also supported for groups)
```

nodeRange tries to be intelligent about detecting padding, so you can specify "node001-node200" and it will add the proper number of zeroes to make all numbers 3 digits.

An incremented range of nodes:

```
node10+3   (equivalent to: node10,node11,node12,node13)
```

A node shorthand range of nodes:

```
10-20    (equivalent to: node10,node11,node12,...node20)
10+3     (equivalent to: node10,node11,node12,node13)
```

Currently, the prefix that will be prepended for the above syntax is always "node". Eventually, the prefix and optional suffix will be settable via the environment variables XCAT_NODE_PREFIX and XCAT_NODE_SUFFIX, but currently this only works in bypass mode.

A regular expression match of nodes or groups:

```
/node[345].*   (will match any nodes that start with node3, node4, or node5)
/group[12].*   (will match any groups that start with group1 or group2)
```

The path of a file containing noderanges of nodes or groups:

```
^/tmp/nodelist
```

where /tmp/nodelist can contain entries like:

```
#my node list (this line ignored)
^/tmp/foo #ignored
node01     #node comment
node02
node03
node10-node20
/group[456].*
-node50
```

Node ranges can contain any combination:

```
node01-node30,node40,^/tmp/nodes,/node[13].*,2-10,node50+5
```

Any individual **noderange** may be prefixed with an exclusion operator (default -) with the exception of the file operator (default ^). This will cause that individual noderange to be subtracted from the total resulting list of nodes.

The intersection operator @ calculates the intersection of the left and right sides:

```
group1@group2    (will result in the list of nodes that group1 and group2 have in
→common)
```

Any combination or multiple combinations of inclusive and exclusive ranges of nodes and groups is legal. There is no precedence implied in the order of the arguments. Exclusive ranges have precedence over inclusive. Parentheses can be used to explicitly specify precedence of any operators.

Nodes have precedence over groups. If a node range match is made then no group range match will be attempted.

All node and group names are validated against the nodelist table. Invalid names are ignored and return nothing.

xCAT Node Name Format

Throughout this man page the term **xCAT Node Name Format** is used. **xCAT Node Name Format** is defined by the following regex:

```
^([A-Za-z-]+)([0-9]+)(([A-Za-z-]+[A-Za-z0-9-]*)*)
```

In plain English, a node or group name is in **xCAT Node Name Format** if starting from the begining there are:

*

   one or more alpha characters of any case and any number of "-" in any combination

*

   followed by one or more numbers

*

   then optionally followed by one alpha character of any case or "-"

*

   followed by any combination of case mixed alphanumerics and "-"

noderange supports node/group names in *any* format. **xCAT Node Name Format** is **not** required, however some node range methods used to determine range will not be used for non-conformant names.

Example of **xCAT Node Name Format** node/group names:

```
NODENAME             PREFIX       NUMBER    SUFFIX
node1                node         1
node001              node         001
node-001             node-        001
node-foo-001-bar     node-foo-    001       -bar
node-foo-1bar        node-foo-    1         bar
foo1bar2             foo          1         bar2
rack01unit34         rack         01        unit34
unit34rack01         unit         34        rack01
pos0134              pos          0134
```

Examples

1. Generates a list of all nodes (assuming all is a group) listed in the **nodelist** table less node5 through node10:

```
all,-node5-node10
```

2. Generates a list of nodes 1 through 10 less nodes 3,4,5. Note that node4 is listed twice, first in the range and then at the end. Because exclusion has precedence node4 will be excluded.

```
node1-node10,-node3-node5,node4
```

3. Generates a list of nodes 1 through 10 less nodes 3 and 5.

```
node1-node10,-node3,-node5
```

4. Generates a list of all (assuming 'all' is a group) nodes in the **nodelist** table less 17 through 32.

```
-node17-node32,all
```

5. Generates a list of nodes 1 through 128, and user nodes 1 through 4.

```
node1-node128,user1-user4
```

6. Generates a list of all nodes (assuming 'all' is a group), less nodes in groups rack1 through rack3 (assuming groups rack1, rack2, and rack3 are defined), less nodes 100 through 200, less nodes in the storage group. Note that node150 is listed but is excluded.

```
all,-rack1-rack3,-node100-node200,node150,-storage
```

7. Generates a list of nodes matching the regex *node[23].**. That is all nodes that start with node2 or node3 and end in anything or nothing. E.g. node2, node3, node20, node30, node21234 all match.

```
/node[23].*
```

8. Generates a list of nodes which have the value hmc in the nodehm.cons attribute.

```
nodehm.cons==hmc

nodehm.cons=~hmc
```

9. Generate a list of nodes in the 1st two frames:

```
f[1-2]n[1-42]
```

SEE ALSO

nodels(1)|nodels.1

man5

auditlog.5

NAME

auditlog - a table in the xCAT database.

SYNOPSIS

auditlog Attributes: *recid*, *audittime*, *userid*, *clientname*, *clienttype*, *command*, *noderange*, *args*, *status*, *comments*, *disable*

DESCRIPTION

```
Audit Data log.
```

auditlog Attributes:

recid

> The record id.

audittime

> The timestamp for the audit entry.

userid

> The user running the command.

clientname

> The client machine, where the command originated.

clienttype

> Type of command: cli,java,webui,other.

command

> Command executed.

noderange

> The noderange on which the command was run.

args

> The command argument list.

status

> Allowed or Denied.

comments

> Any user-provided notes.

disable

> Do not use. tabprune will not work if set to yes or 1

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

bootparams.5

NAME

bootparams - a table in the xCAT database.

SYNOPSIS

bootparams Attributes: *node*, *kernel*, *initrd*, *kcmdline*, *addkcmdline*, *dhcpstatements*, *adddhcpstatements*, *comments*, *disable*

DESCRIPTION

Current boot settings to be sent to systems attempting network boot for deployment, stateless, or other reasons. Mostly automatically manipulated by xCAT.

bootparams Attributes:

node

> The node or group name

kernel

> The kernel that network boot actions should currently acquire and use. Note this could be a chained boot loader such as memdisk or a non-linux boot loader

initrd

> The initial ramdisk image that network boot actions should use (could be a DOS floppy or hard drive image if using memdisk as kernel)

kcmdline

> Arguments to be passed to the kernel

addkcmdline

> User specified one or more parameters to be passed to the kernel. For the kernel options need to be persistent after installation, specify them with prefix "R::"

dhcpstatements

> xCAT manipulated custom dhcp statements (not intended for user manipulation)

adddhcpstatements

> Custom dhcp statements for administrator use (not implemneted yet)

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

boottarget.5

NAME

boottarget - a table in the xCAT database.

SYNOPSIS

boottarget Attributes: *bprofile*, *kernel*, *initrd*, *kcmdline*, *comments*, *disable*

DESCRIPTION

Specify non-standard initrd, kernel, and parameters that should be used for a given profile.

boottarget Attributes:

bprofile

> All nodes with a nodetype.profile value equal to this value and nodetype.os set to "boottarget", will use the associated kernel, initrd, and kcmdline.

kernel

> The kernel that network boot actions should currently acquire and use. Note this could be a chained boot loader such as memdisk or a non-linux boot loader

initrd

> The initial ramdisk image that network boot actions should use (could be a DOS floppy or hard drive image if using memdisk as kernel)

kcmdline

> Arguments to be passed to the kernel

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

cfgmgt.5

NAME

cfgmgt - a table in the xCAT database.

SYNOPSIS

cfgmgt Attributes: *node*, *cfgmgr*, *cfgserver*, *roles*, *comments*, *disable*

DESCRIPTION

Configuration management data for nodes used by non-xCAT osimage management services to install and configure software on a node.

cfgmgt Attributes:

node

> The node being managed by the cfgmgr service

cfgmgr

> The name of the configuration manager service. Currently 'chef' and 'puppet' are supported services.

cfgserver

> The xCAT node name of the chef server or puppet master

roles

> The roles associated with this node as recognized by the cfgmgr for the software that is to be installed and configured. These role names map to chef recipes or puppet manifest classes that should be used for this node. For example, chef OpenStack cookbooks have roles such as mysql-master,keystone, glance, nova-controller, nova-conductor, cinder-all.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

chain.5

NAME

chain - a table in the xCAT database.

SYNOPSIS

chain Attributes: *node*, *currstate*, *currchain*, *chain*, *ondiscover*, *comments*, *disable*

DESCRIPTION

Controls what operations are done (and it what order) when a node is discovered and deployed.

chain Attributes:

node

> The node name or group name.

currstate

> The current or next chain step to be executed on this node by xCAT-genesis. Set by xCAT during node discovery or as a result of nodeset.

currchain

> The chain steps still left to do for this node. This attribute will be automatically adjusted by xCAT while xCAT-genesis is running on the node (either during node discovery or a special operation like firmware update). During node discovery, this attribute is initialized from the chain attribute and updated as the chain steps are executed.

chain

> A comma-delimited chain of actions to be performed automatically when this node is discovered. ("Discovered" means a node booted, but xCAT and DHCP did not recognize the MAC of this node. In this situation, xCAT initiates the discovery process, the last step of which is to run the operations listed in this chain attribute, one by one.) Valid values: boot or reboot, install or netboot, runcmd=<cmd>, runimage=<URL>, shell, standby. (Default - same as no chain - it will do only the discovery.). Example, for BMC machines use: runcmd=bmcsetup,shell.

ondiscover

> This attribute is currently not used by xCAT. The "nodediscover" operation is always done during node discovery.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

deps.5

NAME

deps - a table in the xCAT database.

SYNOPSIS

deps Attributes: *node*, *nodedep*, *msdelay*, *cmd*, *comments*, *disable*

DESCRIPTION

Describes dependencies some nodes have on others. This can be used, e.g., by rpower -d to power nodes on or off in the correct order.

deps Attributes:

node

> The node name or group name.

nodedep

> Comma-separated list of nodes or node groups it is dependent on.

msdelay

> How long to wait between operating on the dependent nodes and the primary nodes.

cmd

> Comma-seperated list of which operation this dependency applies to.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

discoverydata.5

NAME

discoverydata - a table in the xCAT database.

SYNOPSIS

discoverydata Attributes: *uuid, node, method, discoverytime, arch, cpucount, cputype, memory, mtm, serial, nicdriver, nicipv4, nichwaddr, nicpci, nicloc, niconboard, nicfirm, switchname, switchaddr, switchdesc, switchport, otherdata, comments, disable*

DESCRIPTION

Discovery data which sent from genesis.

discoverydata Attributes:

uuid

> The uuid of the node which send out the discovery request.

node

> The node name which assigned to the discovered node.

method

> The method which handled the discovery request. The method could be one of: switch, blade, profile, sequential.

discoverytime

> The last time that xCAT received the discovery message.

arch

> The architecture of the discovered node. e.g. x86_64.

cpucount

> The number of cores multiply by threads core supported for the discovered node. e.g. 192.

cputype

> The cpu type of the discovered node. e.g. Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz

memory

> The memory size of the discovered node. e.g. 198460852

mtm

> The machine type model of the discovered node. e.g. 786310X

serial

> The serial number of the discovered node. e.g. 1052EFB

nicdriver

> The driver of the nic. The value should be comma separated <nic name!driver name>. e.g. eth0!be2net,eth1!be2net

nicipv4

> The ipv4 address of the nic. The value should be comma separated <nic name!ipv4 address>. e.g. eth0!10.0.0.212/8

nichwaddr

The hardware address of the nic. The should will be comma separated <nic name!hardware address>. e.g. eth0!34:40:B5:BE:DB:B0,eth1!34:40:B5:BE:DB:B4

nicpci

The pic device of the nic. The value should be comma separated <nic name!pci device>. e.g. eth0!0000:0c:00.0,eth1!0000:0c:00.1

nicloc

The location of the nic. The value should be comma separated <nic name!nic location>. e.g. eth0!Onboard Ethernet 1,eth1!Onboard Ethernet 2

niconboard

The onboard info of the nic. The value should be comma separated <nic name!onboard info>. e.g. eth0!1,eth1!2

nicfirm

The firmware description of the nic. The value should be comma separated <nic name!fimware description>. e.g. eth0!ServerEngines BE3 Controller,eth1!ServerEngines BE3 Controller

switchname

The switch name which the nic connected to. The value should be comma separated <nic name!switch name>. e.g. eth0!c909f06sw01

switchaddr

The address of the switch which the nic connected to. The value should be comma separated <nic name!switch address>. e.g. eth0!192.168.70.120

switchdesc

The description of the switch which the nic connected to. The value should be comma separated <nic name!switch description>. e.g. eth0!IBM Flex System Fabric EN4093 10Gb Scalable Switch, flash image: version 7.2.6, boot image: version 7.2.6

switchport

The port of the switch that the nic connected to. The value should be comma separated <nic name!switch port>. e.g. eth0!INTA2

otherdata

The left data which is not parsed to specific attributes (The complete message comes from genesis)

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

domain.5

NAME

domain - a table in the xCAT database.

SYNOPSIS

domain Attributes: *node*, *ou*, *authdomain*, *adminuser*, *adminpassword*, *type*, *comments*, *disable*

DESCRIPTION

Mapping of nodes to domain attributes

domain Attributes:

node

> The node or group the entry applies to

ou

> For an LDAP described machine account (i.e. Active Directory), the orginaztional unit to place the system.
> If not set, defaults to cn=Computers,dc=your,dc=domain

authdomain

> If a node should participate in an AD domain or Kerberos realm distinct from domain indicated in site,
> this field can be used to specify that

adminuser

> Allow a node specific indication of Administrative user. Most will want to just use passwd table to indicate
> this once rather than by node.

adminpassword

> Allow a node specific indication of Administrative user password for the domain. Most will want to
> ignore this in favor of passwd table.

type

> Type, if any, of authentication domain to manipulate. The only recognized value at the moment is ac-
> tivedirectory.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

---

eventlog.5

NAME

eventlog - a table in the xCAT database.

SYNOPSIS

eventlog Attributes: *recid*, *eventtime*, *eventtype*, *monitor*, *monnode*, *node*, *application*, *component*, *id*, *severity*, *message*, *rawdata*, *comments*, *disable*

DESCRIPTION

Stores the events occurred.

eventlog Attributes:

recid

> The record id.

eventtime

> The timestamp for the event.

eventtype

> The type of the event.

monitor

> The name of the monitor that monitors this event.

monnode

> The node that monitors this event.

node

> The node where the event occurred.

application

> The application that reports the event.

component

> The component where the event occurred.

id

> The location or the resource name where the event occurred.

severity

> The severity of the event. Valid values are: informational, warning, critical.

message

> The full description of the event.

rawdata

The data that associated with the event.

comments

Any user-provided notes.

disable

Do not use. tabprune will not work if set to yes or 1

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

firmware.5

NAME

firmware - a table in the xCAT database.

SYNOPSIS

firmware Attributes: *node*, *cfgfile*, *comments*, *disable*

DESCRIPTION

Maps node to firmware values to be used for setup at node discovery or later

firmware Attributes:

node

The node id.

cfgfile

The file to use.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

hosts.5

NAME

hosts - a table in the xCAT database.

SYNOPSIS

hosts Attributes: *node*, *ip*, *hostnames*, *otherinterfaces*, *comments*, *disable*

DESCRIPTION

IP addresses and hostnames of nodes. This info is optional and is only used to populate /etc/hosts and DNS via makehosts and makedns. Using regular expressions in this table can be a quick way to populate /etc/hosts.

hosts Attributes:

node

> The node name or group name.

ip

> The IP address of the node. This is only used in makehosts. The rest of xCAT uses system name resolution to resolve node names to IP addresses.

hostnames

> Hostname aliases added to /etc/hosts for this node. Comma or blank separated list.

otherinterfaces

> Other IP addresses to add for this node. Format: -<ext>:<ip>,<intfhostname>:<ip>,...

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

hwinv.5

NAME

hwinv - a table in the xCAT database.

SYNOPSIS

hwinv Attributes: *node*, *cputype*, *cpucount*, *memory*, *disksize*, *comments*, *disable*

DESCRIPTION

The hareware inventory for the node.

hwinv Attributes:

node

>   The node name or group name.

cputype

>   The cpu model name for the node.

cpucount

>   The number of cpus for the node.

memory

>   The size of the memory for the node in MB.

disksize

>   The size of the disks for the node in GB.

comments

>   Any user-provided notes.

disable

>   Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

hypervisor.5

NAME

hypervisor - a table in the xCAT database.

SYNOPSIS

hypervisor Attributes: *node*, *type*, *mgr*, *interface*, *netmap*, *defaultnet*, *cluster*, *datacenter*, *preferdirect*, *comments*, *disable*

DESCRIPTION

Hypervisor parameters

hypervisor Attributes:

node

>   The node or static group name

type

>   The plugin associated with hypervisor specific commands such as revacuate

mgr

>   The virtualization specific manager of this hypervisor when applicable

interface

>   The definition of interfaces for the hypervisor. The format is [network-name:interfacename:bootprotocol:IP:netmask:gateway] that split with | for each interface

netmap

>   Optional mapping of useful names to relevant physical ports. For example, 10ge=vmnic_16.0&vmnic_16.1,ge=vmnic1 would be requesting two virtual switches to be created, one called 10ge with vmnic_16.0 and vmnic_16.1 bonded, and another simply connected to vmnic1. Use of this allows abstracting guests from network differences amongst hypervisors

defaultnet

>   Optionally specify a default network entity for guests to join to if they do not specify.

cluster

>   Specify to the underlying virtualization infrastructure a cluster membership for the hypervisor.

datacenter

>   Optionally specify a datacenter for the hypervisor to exist in (only applicable to VMWare)

preferdirect

>   If a mgr is declared for a hypervisor, xCAT will default to using the mgr for all operations. If this is field is set to yes or 1, xCAT will prefer to directly communicate with the hypervisor if possible

comments

disable

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

ipmi.5

NAME

ipmi - a table in the xCAT database.

SYNOPSIS

ipmi Attributes: *node*, *bmc*, *bmcport*, *taggedvlan*, *bmcid*, *username*, *password*, *comments*, *disable*

DESCRIPTION

Settings for nodes that are controlled by an on-board BMC via IPMI.

ipmi Attributes:

node

> The node name or group name.

bmc

> The hostname of the BMC adapater.

bmcport

```
In systems with selectable shared/dedicated ethernet ports,
        this parameter can be used to specify the preferred port.  0
        means use the shared port, 1 means dedicated, blank is to not
        assign.

        The following special cases exist for IBM System x servers:

        For x3755 M3 systems, 0 means use the dedicated port, 1 means
        shared, blank is to not assign.

    For certain systems which have a mezzaine or ML2 adapter, there is a␣
→second
    value to include:


        For x3750 M4 (Model 8722):


        0 2   1st 1Gbps interface for LOM

        0 0   1st 10Gbps interface for LOM

        0 3   2nd 1Gbps interface for LOM

        0 1   2nd 10Gbps interface for LOM


        For  x3750 M4 (Model 8752), x3850/3950 X6, dx360 M4, x3550 M4, and␣
→x3650 M4:


        0     Shared (1st onboard interface)

        1     Dedicated

        2 0   First interface on ML2 or mezzanine adapter
```

```
        2 1    Second interface on ML2 or mezzanine adapter

        2 2    Third interface on ML2 or mezzanine adapter

        2 3    Fourth interface on ML2 or mezzanine adapter
```

taggedvlan

> Have bmcsetup place the BMC on the specified vlan tag on a shared netwirk interface. Some network devices may be incompatible with this option

bmcid

> Unique identified data used by discovery processes to distinguish known BMCs from unrecognized BMCs

username

> The BMC userid. If not specified, the key=ipmi row in the passwd table is used as the default.

password

> The BMC password. If not specified, the key=ipmi row in the passwd table is used as the default.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

iscsi.5

NAME

iscsi - a table in the xCAT database.

SYNOPSIS

iscsi Attributes: *node*, *server*, *target*, *lun*, *iname*, *file*, *userid*, *passwd*, *kernel*, *kcmdline*, *initrd*, *comments*, *disable*

DESCRIPTION

Contains settings that control how to boot a node from an iSCSI target

iscsi Attributes:

node

> The node name or group name.

---

server

> The server containing the iscsi boot device for this node.

target

> The iscsi disk used for the boot device for this node. Filled in by xCAT.

lun

> LUN of boot device. Per RFC-4173, this is presumed to be 0 if unset. tgtd often requires this to be 1

iname

> Initiator name. Currently unused.

file

> The path on the server of the OS image the node should boot from.

userid

> The userid of the iscsi server containing the boot device for this node.

passwd

> The password for the iscsi server containing the boot device for this node.

kernel

> The path of the linux kernel to boot from.

kcmdline

> The kernel command line to use with iSCSI for this node.

initrd

> The initial ramdisk to use when network booting this node.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

kit.5

NAME

kit - a table in the xCAT database.

SYNOPSIS

kit Attributes: *kitname*, *basename*, *description*, *version*, *release*, *ostype*, *isinternal*, *kitdeployparams*, *kitdir*, *comments*, *disable*

DESCRIPTION

This table stores all kits added to the xCAT cluster.

kit Attributes:

kitname

The unique generated kit name, when kit is added to the cluster.

basename

The kit base name

description

The Kit description.

version

The kit version

release

The kit release

ostype

The kit OS type. Linux or AIX.

isinternal

A flag to indicated if the Kit is internally used. When set to 1, the Kit is internal. If 0 or undefined, the kit is not internal.

kitdeployparams

The file containing the default deployment parameters for this Kit. These parameters are added to the OS Image definition.s list of deployment parameters when one or more Kit Components from this Kit are added to the OS Image.

kitdir

The path to Kit Installation directory on the Mgt Node.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

kitcomponent.5

NAME

kitcomponent - a table in the xCAT database.

SYNOPSIS

kitcomponent Attributes: *kitcompname*, *description*, *kitname*, *kitreponame*, *basename*, *version*, *release*, *serverroles*, *kitpkgdeps*, *prerequisite*, *driverpacks*, *kitcompdeps*, *postbootscripts*, *genimage_postinstall*, *exlist*, *comments*, *disable*

DESCRIPTION

This table stores all kit components added to the xCAT cluster.

kitcomponent Attributes:

kitcompname

   The unique Kit Component name. It is auto-generated when the parent Kit is added to the cluster.

description

   The Kit component description.

kitname

   The Kit name which this Kit Component belongs to.

kitreponame

   The Kit Package Repository name which this Kit Component belongs to.

basename

   Kit Component basename.

version

   Kit Component version.

release

   Kit Component release.

serverroles

   The types of servers that this Kit Component can install on. Valid types are: mgtnode, servicenode, compute

kitpkgdeps

   Comma-separated list of packages that this kit component depends on.

prerequisite

   Prerequisite for this kit component, the prerequisite includes ospkgdeps,preinstall,preupgrade,preuninstall scripts

driverpacks

Comma-separated List of driver package names. These must be full names like: pkg1-1.0-1.x86_64.rpm.

kitcompdeps

Comma-separated list of kit components that this kit component depends on.

postbootscripts

Comma-separated list of postbootscripts that will run during the node boot.

genimage_postinstall

Comma-separated list of postinstall scripts that will run during the genimage.

exlist

Exclude list file containing the files/directories to exclude when building a diskless image.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

kitrepo.5

NAME

kitrepo - a table in the xCAT database.

SYNOPSIS

kitrepo Attributes: *kitreponame*, *kitname*, *osbasename*, *osmajorversion*, *osminorversion*, *osarch*, *compat_osbasenames*, *kitrepodir*, *comments*, *disable*

DESCRIPTION

This table stores all kits added to the xCAT cluster.

kitrepo Attributes:

kitreponame

The unique generated kit repo package name, when kit is added to the cluster.

kitname

The Kit name which this Kit Package Repository belongs to.

osbasename

> The OS distro name which this repository is based on.

osmajorversion

> The OS distro major version which this repository is based on.

osminorversion

> The OS distro minor version which this repository is based on. If this attribute is not set, it means that this repo applies to all minor versions.

osarch

> The OS distro arch which this repository is based on.

compat_osbasenames

> List of compatible OS base names.

kitrepodir

> The path to Kit Repository directory on the Mgt Node.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

kvm_masterdata.5

NAME

kvm_masterdata - a table in the xCAT database.

SYNOPSIS

kvm_masterdata Attributes: *name*, *xml*, *comments*, *disable*

DESCRIPTION

Persistant store for KVM plugin for masters

kvm_masterdata Attributes:

name

> The name of the relevant master

xml

The XML description to be customized for clones of this master

comments

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

kvm_nodedata.5

NAME

kvm_nodedata - a table in the xCAT database.

SYNOPSIS

kvm_nodedata Attributes: *node*, *xml*, *comments*, *disable*

DESCRIPTION

Persistant store for KVM plugin, not intended for manual modification.

kvm_nodedata Attributes:

node

The node corresponding to the virtual machine

xml

The XML description generated by xCAT, fleshed out by libvirt, and stored for reuse

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

linuximage.5

NAME

linuximage - a table in the xCAT database.

SYNOPSIS

linuximage Attributes: *imagename*, *template*, *boottarget*, *addkcmdline*, *pkglist*, *pkgdir*, *otherpkglist*, *otherpkgdir*, *exlist*, *postinstall*, *rootimgdir*, *kerneldir*, *nodebootif*, *otherifce*, *netdrivers*, *kernelver*, *krpmver*, *permission*, *dump*, *crashkernelsize*, *partitionfile*, *driverupdatesrc*, *comments*, *disable*

DESCRIPTION

Information about a Linux operating system image that can be used to deploy cluster nodes.

linuximage Attributes:

imagename

> The name of this xCAT OS image definition.

template

> The fully qualified name of the template file that will be used to create the OS installer configuration file for stateful installations (e.g. kickstart for RedHat, autoyast for SLES).

boottarget

> The name of the boottarget definition. When this attribute is set, xCAT will use the kernel, initrd and kernel params defined in the boottarget definition instead of the default.

addkcmdline

> User specified arguments to be passed to the kernel. The user arguments are appended to xCAT.s default kernel arguments. For the kernel options need to be persistent after installation, specify them with prefix "R::". This attribute is ignored if linuximage.boottarget is set.

pkglist

> The fully qualified name of the file that stores the distro packages list that will be included in the image. Make sure that if the pkgs in the pkglist have dependency pkgs, the dependency pkgs should be found in one of the pkgdir

pkgdir

> The name of the directory where the distro packages are stored. It could be set multiple paths.The multiple paths must be seperated by ",". The first path in the value of osimage.pkgdir must be the OS base pkg dir path, such as pkgdir=/install/rhels6.2/x86_64,/install/updates . In the os base pkg path, there are default repository data. And in the other pkg path(s), the users should make sure there are repository data. If not, use "createrepo" command to create them. For ubuntu, multiple mirrors can be specified in the pkgdir attribute, the mirrors must be prefixed by the protocol(http/ssh) and delimited with "," between each other.

otherpkglist

> The fully qualified name of the file that stores non-distro package lists that will be included in the image. It could be set multiple paths.The multiple paths must be seperated by ",".

---

otherpkgdir

> The base directory where the non-distro packages are stored. Only 1 local directory supported at present.

exlist

> The fully qualified name of the file that stores the file names and directory names that will be excluded from the image during packimage command. It is used for diskless image only.

postinstall

> The fully qualified name of the script file that will be run at the end of the genimage command. It could be set multiple paths.The multiple paths must be seperated by ",". It is used for diskless image only.

rootimgdir

> The directory name where the image is stored. It is generally used for diskless image. it also can be used in sysclone environment to specify where the image captured from golden client is stored. in sysclone environment, rootimgdir is generally assigned to some default value by xcat, but you can specify your own store directory. just one thing need to be noticed, wherever you save the image, the name of last level directory must be the name of image. for example, if your image name is testimage and you want to save this image under home directoy, rootimgdir should be assigned to value /home/testimage/

kerneldir

> The directory name where the 3rd-party kernel is stored. It is used for diskless image only.

nodebootif

> The network interface the stateless/statelite node will boot over (e.g. eth0)

otherifce

> Other network interfaces (e.g. eth1) in the image that should be configured via DHCP

netdrivers

> The ethernet device drivers of the nodes which will use this linux image, at least the device driver for the nodes' installnic should be included

kernelver

> The version of linux kernel used in the linux image. If the kernel version is not set, the default kernel in rootimgdir will be used

krpmver

> The rpm version of kernel packages (for SLES only). If it is not set, the default rpm version of kernel packages will be used.

permission

> The mount permission of /.statelite directory is used, its default value is 755

dump

> The NFS directory to hold the Linux kernel dump file (vmcore) when the node with this image crashes, its format is "nfs://<nfs_server_ip>/<kdump_path>". If you want to use the node's "xcatmaster" (its SN or MN), <nfs_server_ip> can be left blank. For example, "nfs:///<kdump_path>" means the NFS directory to hold the kernel dump file is on the node's SN, or MN if there's no SN.

crashkernelsize

> the size that assigned to the kdump kernel. If the kernel size is not set, 256M will be the default value.

partitionfile

The path of the configuration file which will be used to partition the disk for the node. For stateful os-images,two types of files are supported: "<partition file absolute path>" which contains a partitioning definition that will be inserted directly into the generated autoinst configuration file and must be formatted for the corresponding OS installer (e.g. kickstart for RedHat, autoyast for SLES, pressed for Ubuntu). "s:<partitioning script absolute path>" which specifies a shell script that will be run from the OS installer configuration file %pre section; the script must write the correct partitioning definition into the file /tmp/partitionfile on the node which will be included into the configuration file during the install process. For statelite osimages, partitionfile should specify "<partition file absolute path>"; see the xCAT Statelite documentation for the xCAT defined format of this configuration file.For Ubuntu, besides "<partition file absolute path>" or "s:<partitioning script absolute path>", the disk name(s) to partition must be specified in traditional, non-devfs format, delimited with space, it can be specified in 2 forms: "d:<the absolute path of the disk name file>" which contains the disk name(s) to partition and "s:d:<the absolute path of the disk script>" which runs in pressed/early_command and writes the disk names into the "/tmp/boot_disk" . To support other specific partition methods such as RAID or LVM in Ubuntu, some additional preseed values should be specified, these values can be specified with "c:<the absolute path of the additional pressed config file>" which contains the additional pressed entries in "d-i ..." form and "s:c:<the absolute path of the additional pressed config script>" which runs in pressed/early_command and set the preseed values with "debconf-set". The multiple values should be delimited with comma ","

driverupdatesrc

The source of the drivers which need to be loaded during the boot. Two types of driver update source are supported: Driver update disk and Driver rpm package. The value for this attribute should be comma separated sources. Each source should be the format tab:full_path_of_srouce_file. The tab keyword can be: dud (for Driver update disk) and rpm (for driver rpm). If missing the tab, the rpm format is the default. e.g. dud:/install/dud/dd.img,rpm:/install/rpm/d.rpm

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

litefile.5

NAME

litefile - a table in the xCAT database.

SYNOPSIS

litefile Attributes: *image*, *file*, *options*, *comments*, *disable*

DESCRIPTION

The litefile table specifies the directories and files on the statelite nodes that should be readwrite, persistent, or readonly overlay. All other files in the statelite nodes come from the readonly statelite image.

litefile Attributes:

image

The name of the image (as specified in the osimage table) that will use these options on this dir/file. You can also specify an image group name that is listed in the osimage.groups attribute of some osimages. 'ALL' means use this row for all images.

file

The full pathname of the file. e.g: /etc/hosts. If the path is a directory, then it should be terminated with a '/'.

options

Options for the file:

```
tmpfs - It is the default option if you leave the options column blank. It␣
→provides a file or directory for the node to use when booting, its␣
→permission will be the same as the original version on the server. In most␣
→cases, it is read-write; however, on the next statelite boot, the original␣
→version of the file or directory on the server will be used, it means it␣
→is non-persistent. This option can be performed on files and directories..

rw - Same as Above.Its name "rw" does NOT mean it always be read-write, even␣
→in most cases it is read-write. Please do not confuse it with the "rw"␣
→permission in the file system.

persistent - It provides a mounted file or directory that is copied to the␣
→xCAT persistent location and then over-mounted on the local file or␣
→directory. Anything written to that file or directory is preserved. It␣
→means, if the file/directory does not exist at first, it will be copied to␣
→the persistent location. Next time the file/directory in the persistent␣
→location will be used. The file/directory will be persistent across␣
→reboots. Its permission will be the same as the original one in the␣
→statelite location. It requires the statelite table to be filled out with␣
→a spot for persistent statelite. This option can be performed on files and␣
→directories.

con - The contents of the pathname are concatenated to the contents of the␣
→existing file. For this directive the searching in the litetree hierarchy␣
→does not stop when the first match is found. All files found in the␣
→hierarchy will be concatenated to the file when found. The permission of␣
→the file will be "-rw-r--r--", which means it is read-write for the root␣
→user, but readonly for the others. It is non-persistent, when the node␣
→reboots, all changes to the file will be lost. It can only be performed on␣
→files. Please do not use it for one directory.

ro - The file/directory will be overmounted read-only on the local file/
→directory. It will be located in the directory hierarchy specified in the␣
→litetree table. Changes made to this file or directory on the server will␣
→be immediately seen in this file/directory on the node. This option␣
→requires that the file/directory to be mounted must be available in one of␣
→the entries in the litetree table. This option can be performed on files␣
→and directories.

link - It provides one file/directory for the node to use when booting, it␣
→is copied from the server, and will be placed in tmpfs on the booted node.␣
→In the local file system of the booted node, it is one symbolic link to␣
→one file/directory in tmpfs. And the permission of the symbolic link is␣
→"lrwxrwxrwx", which is not the real permission of the file/directory on␣
→the node. So for some application sensitive to file permissions, it will␣
→be one issue to use "link" as its option, for example, ssh, the keys␣
→which is used for SSH, should NOT use "link" as its option. It is non-
→persistent, when the node is rebooted, all changes to the file/directory␣
→will be lost. This option can be performed on files and directories.
```

```
link,con -  It works similar to the "con" option. All the files found in the␣
→litetree hierarchy will be concatenated to the file when found. The final␣
→file will be put to the tmpfs on the booted node. In the local file system␣
→of the booted node, it is one symbolic link to the file/directory in tmpfs.
→ It is non-persistent, when the node is rebooted, all changes to the file␣
→will be lost. The option can only be performed on files.

 link,persistent - It provides a mounted file or directory that is copied to␣
→the xCAT persistent location and then over-mounted to the tmpfs on the␣
→booted node, and finally the symbolic link in the local file system will␣
→be linked to the over-mounted tmpfs file/directory on the booted node. The␣
→file/directory will be persistent across reboots. The permission of the␣
→file/directory where the symbolic link points to will be the same as the␣
→original one in the statelite location. It requires the statelite table to␣
→be filled out with a spot for persistent statelite. The option can be␣
→performed on files and directories.

link,ro - The file is readonly, and will be placed in tmpfs on the booted␣
→node. In the local file system of the booted node, it is one symbolic link␣
→to the tmpfs. It is non-persistent, when the node is rebooted, all changes␣
→to the file/directory will be lost. This option requires that the file/
→directory to be mounted must be available in one of the entries in the␣
→litetree table. The option can be performed on files and directories.
```

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

litetree.5

NAME

litetree - a table in the xCAT database.

SYNOPSIS

litetree Attributes: *priority*, *image*, *directory*, *mntopts*, *comments*, *disable*

DESCRIPTION

Directory hierarchy to traverse to get the initial contents of node files. The files that are specified in the litefile table are searched for in the directories specified in this table.

litetree Attributes:

priority

> This number controls what order the directories are searched. Directories are searched from smallest priority number to largest.

image

> The name of the image (as specified in the osimage table) that will use this directory. You can also specify an image group name that is listed in the osimage.groups attribute of some osimages. 'ALL' means use this row for all images.

directory

> The location (hostname:path) of a directory that contains files specified in the litefile table. Variables are allowed. E.g: $noderes.nfsserver://xcatmasternode/install/$node/#CMD=uname-r#/

mntopts

> A comma-separated list of options to use when mounting the litetree directory. (Ex. 'soft') The default is to do a 'hard' mount.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

mac.5

NAME

mac - a table in the xCAT database.

SYNOPSIS

mac Attributes: *node*, *interface*, *mac*, *comments*, *disable*

DESCRIPTION

The MAC address of the node's install adapter. Normally this table is populated by getmacs or node discovery, but you can also add entries to it manually.

mac Attributes:

node

> The node name or group name.

interface

> The adapter interface name that will be used to install and manage the node. E.g. eth0 (for linux) or en0 (for AIX).)

mac

> The mac address or addresses for which xCAT will manage static bindings for this node. This may be simply a mac address, which would be bound to the node name (such as "01:02:03:04:05:0E"). This may also be a "|" delimited string of "mac address!hostname" format (such as "01:02:03:04:05:0E!node5|01:02:03:05:0F!node6-eth1").

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

mic.5

NAME

mic - a table in the xCAT database.

SYNOPSIS

mic Attributes: *node*, *host*, *id*, *nodetype*, *bridge*, *onboot*, *vlog*, *powermgt*, *comments*, *disable*

DESCRIPTION

The host, slot id and configuraton of the mic (Many Integrated Core).

mic Attributes:

node

> The node name or group name.

host

> The host node which the mic card installed on.

id

The device id of the mic node.

nodetype

The hardware type of the mic node. Generally, it is mic.

bridge

The virtual bridge on the host node which the mic connected to.

onboot

Set mic to autoboot when mpss start. Valid values: yes|no. Default is yes.

vlog

Set the Verbose Log to console. Valid values: yes|no. Default is no.

powermgt

Set the Power Management for mic node. This attribute is used to set the power management state that mic may get into when it is idle. Four states can be set: cpufreq, corec6, pc3 and pc6. The valid value for powermgt attribute should be [cpufreq=<on|off>]![corec6=<on|off>]![pc3=<on|off>]![pc6=<on|off>]. e.g. cpufreq=on!corec6=off!pc3=on!pc6=off. Refer to the doc of mic to get more information for power management.

comments

Any user-provided notes.

disable

Do not use. tabprune will not work if set to yes or 1

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

monitoring.5

NAME

monitoring - a table in the xCAT database.

SYNOPSIS

monitoring Attributes: *name*, *nodestatmon*, *comments*, *disable*

DESCRIPTION

Controls what external monitoring tools xCAT sets up and uses. Entries should be added and removed from this table using the provided xCAT commands monstart and monstop.

monitoring Attributes:

name

> The name of the mornitoring plug-in module. The plug-in must be put in /lib/perl/xCAT_monitoring/. See the man page for monstart for details.

nodestatmon

> Specifies if the monitoring plug-in is used to feed the node status to the xCAT cluster. Any one of the following values indicates "yes": y, Y, yes, Yes, YES, 1. Any other value or blank (default), indicates "no".

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

monsetting.5

NAME

monsetting - a table in the xCAT database.

SYNOPSIS

monsetting Attributes: *name*, *key*, *value*, *comments*, *disable*

DESCRIPTION

Specifies the monitoring plug-in specific settings. These settings will be used by the monitoring plug-in to customize the behavior such as event filter, sample interval, responses etc. Entries should be added, removed or modified by chtab command. Entries can also be added or modified by the monstart command when a monitoring plug-in is brought up.

monsetting Attributes:

name

> The name of the mornitoring plug-in module. The plug-in must be put in /lib/perl/xCAT_monitoring/. See the man page for monstart for details.

key

> Specifies the name of the attribute. The valid values are specified by each monitoring plug-in. Use "monls name -d" to get a list of valid keys.

value

Specifies the value of the attribute.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

mp.5

NAME

mp - a table in the xCAT database.

SYNOPSIS

mp Attributes: *node*, *mpa*, *id*, *nodetype*, *comments*, *disable*

DESCRIPTION

Contains the hardware control info specific to blades. This table also refers to the mpa table, which contains info about each Management Module.

mp Attributes:

node

The blade node name or group name.

mpa

The managment module used to control this blade.

id

The slot number of this blade in the BladeCenter chassis.

nodetype

The hardware type for mp node. Valid values: mm,cmm, blade.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

mpa.5

NAME

mpa - a table in the xCAT database.

SYNOPSIS

mpa Attributes: *mpa*, *username*, *password*, *displayname*, *slots*, *urlpath*, *comments*, *disable*

DESCRIPTION

Contains info about each Management Module and how to access it.

mpa Attributes:

mpa

> Hostname of the management module.

username

> Userid to use to access the management module.

password

> Password to use to access the management module. If not specified, the key=blade row in the passwd table is used as the default.

displayname

> Alternative name for BladeCenter chassis. Only used by PCM.

slots

> **The number of available slots in the chassis. For PCM, this attribute is used to store the number of slots in the following fo**
> > \<slot rows\> = number of rows of slots in chassis \<slot columns\> = number of columns of slots in chassis \<slot orientation\> = set to 0 if slots are vertical, and set to 1 if slots of horizontal

urlpath

> URL path for the Chassis web interface. The full URL is built as follows: \<hostname\>/\<urlpath\>

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

networks.5

NAME

networks - a table in the xCAT database.

SYNOPSIS

networks Attributes: *netname*, *net*, *mask*, *mgtifname*, *gateway*, *dhcpserver*, *tftpserver*, *nameservers*, *ntpservers*, *logservers*, *dynamicrange*, *staticrange*, *staticrangeincrement*, *nodehostname*, *ddnsdomain*, *vlanid*, *domain*, *comments*, *disable*

DESCRIPTION

Describes the networks in the cluster and info necessary to set up nodes on that network.

networks Attributes:

netname

> Name used to identify this network definition.

net

> The network address.

mask

> The network mask.

mgtifname

> The interface name of the management/service node facing this network. !remote!<nicname> indicates a non-local network on a specific nic for relay DHCP.

gateway

> The network gateway. It can be set to an ip address or the keyword <xcatmaster>, the keyword <xcatmaster> indicates the cluster-facing ip address configured on this management node or service node. Leaving this field blank means that there is no gateway for this network.

dhcpserver

> The DHCP server that is servicing this network. Required to be explicitly set for pooled service node operation.

tftpserver

> The TFTP server that is servicing this network. If not set, the DHCP server is assumed.

nameservers

A comma delimited list of DNS servers that each node in this network should use. This value will end up in the nameserver settings of the /etc/resolv.conf on each node in this network. If this attribute value is set to the IP address of an xCAT node, make sure DNS is running on it. In a hierarchical cluster, you can also set this attribute to "<xcatmaster>" to mean the DNS server for each node in this network should be the node that is managing it (either its service node or the management node). Used in creating the DHCP network definition, and DNS configuration.

ntpservers

The ntp servers for this network. Used in creating the DHCP network definition. Assumed to be the DHCP server if not set.

logservers

The log servers for this network. Used in creating the DHCP network definition. Assumed to be the DHCP server if not set.

dynamicrange

The IP address range used by DHCP to assign dynamic IP addresses for requests on this network. This should not overlap with entities expected to be configured with static host declarations, i.e. anything ever expected to be a node with an address registered in the mac table.

staticrange

The IP address range used to dynamically assign static IPs to newly discovered nodes. This should not overlap with the dynamicrange nor overlap with entities that were manually assigned static IPs. The format for the attribute value is: <startip>-<endip>.

staticrangeincrement

nodehostname

A regular expression used to specify node name to network-specific hostname. i.e. "/z/-secondary/" would mean that the hostname of "n1" would be n1-secondary on this network. By default, the nodename is assumed to equal the hostname, followed by nodename-interfacename.

ddnsdomain

A domain to be combined with nodename to construct FQDN for DDNS updates induced by DHCP. This is not passed down to the client as "domain"

vlanid

The vlan ID if this network is within a vlan.

domain

The DNS domain name (ex. cluster.com).

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

nics.5

NAME

nics - a table in the xCAT database.

SYNOPSIS

nics Attributes: *node*, *nicips*, *nichostnamesuffixes*, *nichostnameprefixes*, *nictypes*, *niccustomscripts*, *nicnetworks*, *nicaliases*, *nicextraparams*, *comments*, *disable*

DESCRIPTION

Stores NIC details.

nics Attributes:

node

> The node or group name.

nicips

> **Comma-separated list of IP addresses per NIC. To specify one ip address per NIC:**
>
> > <nic1>!<ip1>,<nic2>!<ip2>,..., for example, eth0!10.0.0.100,ib0!11.0.0.100
>
> **To specify multiple ip addresses per NIC:** <nic1>!<ip1>|<ip2>,<nic2>!<ip1>|<ip2>,..., for example, eth0!10.0.0.100|fd55::214:5eff:fe15:849b,ib0!11.0.0.100|2001::214:5eff:fe15:849a. The xCAT object definition commands support to use nicips.<nicname> as the sub attributes.
>
> Note: The primary IP address must also be stored in the hosts.ip attribute. The nichostnamesuffixes should specify one hostname suffix for each ip address.

nichostnamesuffixes

> **Comma-separated list of hostname suffixes per NIC.**
>
> **If only one ip address is associated with each NIC:** <nic1>!<ext1>,<nic2>!<ext2>,..., for example, eth0!-eth0,ib0!-ib0
>
> **If multiple ip addresses are assocciated with each NIC:** <nic1>!<ext1>|<ext2>,<nic2>!<ext1>|<ext2>,..., for example, eth0!-eth0|-eth0-ipv6,ib0!-ib0|-ib0-ipv6.
>
> The xCAT object definition commands support to use nichostnamesuffixes.<nicname> as the sub attributes. Note: According to DNS rules a hostname must be a text string up to 24 characters drawn from the alphabet (A-Z), digits (0-9), minus sign (-),and period (.). When you are specifying "nichostnamesuffixes" or "nicaliases" make sure the resulting hostnames will conform to this naming convention

nichostnameprefixes

> **Comma-separated list of hostname prefixes per NIC.**
>
> **If only one ip address is associated with each NIC:** <nic1>!<ext1>,<nic2>!<ext2>,..., for example, eth0!eth0-,ib0!ib-

> **If multiple ip addresses are associated with each NIC:** <nic1>!<ext1>|<ext2>,<nic2>!<ext1>|<ext2>,...,
> for example, eth0!eth0-|eth0-ipv6i-,ib0!ib-|ib-ipv6-.

The xCAT object definition commands support to use nichostnameprefixes.<nicname> as the sub attributes. Note: According to DNS rules a hostname must be a text string up to 24 characters drawn from the alphabet (A-Z), digits (0-9), minus sign (-),and period (.). When you are specifying "nichostnameprefixes" or "nicaliases" make sure the resulting hostnames will conform to this naming convention

nictypes

Comma-separated list of NIC types per NIC. <nic1>!<type1>,<nic2>!<type2>, e.g. eth0!Ethernet,ib0!Infiniband. The xCAT object definition commands support to use nictypes.<nicname> as the sub attributes.

niccustomscripts

Comma-separated list of custom scripts per NIC. <nic1>!<script1>,<nic2>!<script2>, e.g. eth0!configeth eth0, ib0!configib ib0. The xCAT object definition commands support to use niccustomscripts.<nicname> as the sub attribute .

nicnetworks

Comma-separated list of networks connected to each NIC.

> **If only one ip address is associated with each NIC:** <nic1>!<network1>,<nic2>!<network2>,
> for example, eth0!10_0_0_0-255_255_0_0, ib0!11_0_0_0-255_255_0_0

> **If multiple ip addresses are associated with each NIC:** <nic1>!<network1>|<network2>,<nic2>!<network1>|<network2>,
> for example, eth0!10_0_0_0-255_255_0_0|fd55:faaf:e1ab:336::/64,ib0!11_0_0_0-255_255_0_0|2001:db8:1:0::/64. The xCAT object definition commands support to use nicnetworks.<nicname> as the sub attributes.

nicaliases

Comma-separated list of hostname aliases for each NIC.

> **Format: eth0!<alias list>,eth1!<alias1 list>|<alias2 list>** For multiple aliases per nic use a space-separated list.

> For example: eth0!moe larry curly,eth1!tom|jerry

nicextraparams

Comma-separated list of extra parameters that will be used for each NIC configuration.

> **If only one ip address is associated with each NIC:** <nic1>!<param1=value1
> param2=value2>,<nic2>!<param3=value3>, for example,
> eth0!MTU=1500,ib0!MTU=65520 CONNECTED_MODE=yes.

> **If multiple ip addresses are associated with each NIC:** <nic1>!<param1=value1
> param2=value2>|<param3=value3>,<nic2>!<param4=value4
> param5=value5>|<param6=value6>, for example,
> eth0!MTU=1500|MTU=1460,ib0!MTU=65520 CONNECTED_MODE=yes.

The xCAT object definition commands support to use nicextraparams.<nicname> as the sub attributes.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

nimimage.5

NAME

nimimage - a table in the xCAT database.

SYNOPSIS

nimimage Attributes: *imagename*, *nimtype*, *lpp_source*, *spot*, *root*, *dump*, *paging*, *resolv_conf*, *tmp*, *home*, *shared_home*, *res_group*, *nimmethod*, *script*, *bosinst_data*, *installp_bundle*, *mksysb*, *fb_script*, *shared_root*, *otherp-kgs*, *image_data*, *configdump*, *comments*, *disable*

DESCRIPTION

All the info that specifies a particular AIX operating system image that can be used to deploy AIX nodes.

nimimage Attributes:

imagename

>   User provided name of this xCAT OS image definition.

nimtype

>   The NIM client type- standalone, diskless, or dataless.

lpp_source

>   The name of the NIM lpp_source resource.

spot

>   The name of the NIM SPOT resource.

root

>   The name of the NIM root resource.

dump

>   The name of the NIM dump resource.

paging

>   The name of the NIM paging resource.

resolv_conf

>   The name of the NIM resolv_conf resource.

tmp

>   The name of the NIM tmp resource.

home

> The name of the NIM home resource.

shared_home

> The name of the NIM shared_home resource.

res_group

> The name of a NIM resource group.

nimmethod

> The NIM install method to use, (ex. rte, mksysb).

script

> The name of a NIM script resource.

bosinst_data

> The name of a NIM bosinst_data resource.

installp_bundle

> One or more comma separated NIM installp_bundle resources.

mksysb

> The name of a NIM mksysb resource.

fb_script

> The name of a NIM fb_script resource.

shared_root

> A shared_root resource represents a directory that can be used as a / (root) directory by one or more diskless clients.

otherpkgs

> One or more comma separated installp or rpm packages. The rpm packages must have a prefix of 'R:', (ex. R:foo.rpm)

image_data

> The name of a NIM image_data resource.

configdump

> Specifies the type of system dump to be collected. The values are selective, full, and none. The default is selective.

comments

> Any user-provided notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

nodegroup.5

NAME

nodegroup - a table in the xCAT database.

SYNOPSIS

nodegroup Attributes: *groupname*, *grouptype*, *members*, *membergroups*, *wherevals*, *comments*, *disable*

DESCRIPTION

Contains group definitions, whose membership is dynamic depending on characteristics of the node.

nodegroup Attributes:

groupname

> Name of the group.

grouptype

> The only current valid value is dynamic. We will be looking at having the object def commands working with static group definitions in the nodelist table.

members

> The value of the attribute is not used, but the attribute is necessary as a place holder for the object def commands. (The membership for static groups is stored in the nodelist table.)

membergroups

> This attribute stores a comma-separated list of nodegroups that this nodegroup refers to. This attribute is only used by PCM.

wherevals

> A list of "attr*val" pairs that can be used to determine the members of a dynamic group, the delimiter is "::" and the operator * can be ==, =~, != or !~.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

nodehm.5

NAME

nodehm - a table in the xCAT database.

SYNOPSIS

nodehm Attributes: *node*, *power*, *mgt*, *cons*, *termserver*, *termport*, *conserver*, *serialport*, *serialspeed*, *serialflow*, *getmac*, *cmdmapping*, *consoleondemand*, *comments*, *disable*

DESCRIPTION

Settings that control how each node's hardware is managed. Typically, an additional table that is specific to the hardware type of the node contains additional info. E.g. the ipmi, mp, and ppc tables.

nodehm Attributes:

node

> The node name or group name.

power

> The method to use to control the power of the node. If not set, the mgt attribute will be used. Valid values: ipmi, blade, hmc, ivm, fsp, kvm, esx, rhevm. If "ipmi", xCAT will search for this node in the ipmi table for more info. If "blade", xCAT will search for this node in the mp table. If "hmc", "ivm", or "fsp", xCAT will search for this node in the ppc table.

mgt

> The method to use to do general hardware management of the node. This attribute is used as the default if power or getmac is not set. Valid values: ipmi, blade, hmc, ivm, fsp, bpa, kvm, esx, rhevm. See the power attribute for more details.

cons

> The console method. If nodehm.serialport is set, this will default to the nodehm.mgt setting, otherwise it defaults to unused. Valid values: cyclades, mrv, or the values valid for the mgt attribute.

termserver

> The hostname of the terminal server.

termport

> The port number on the terminal server that this node is connected to.

conserver

> The hostname of the machine where the conserver daemon is running. If not set, the default is the xCAT management node.

serialport

> The serial port for this node, in the linux numbering style (0=COM1/ttyS0, 1=COM2/ttyS1). For SOL on IBM blades, this is typically 1. For rackmount IBM servers, this is typically 0.

serialspeed

> The speed of the serial port for this node. For SOL this is typically 19200.

serialflow

> The flow control value of the serial port for this node. For SOL this is typically 'hard'.

getmac

> The method to use to get MAC address of the node with the getmac command. If not set, the mgt attribute will be used. Valid values: same as values for mgmt attribute.

cmdmapping

> The fully qualified name of the file that stores the mapping between PCM hardware management commands and xCAT/third-party hardware management commands for a particular type of hardware device. Only used by PCM.

consoleondemand

> This overrides the value from site.consoleondemand; (0=no, 1=yes). Default is the result from site.consoleondemand.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

nodelist.5

NAME

nodelist - a table in the xCAT database.

SYNOPSIS

nodelist Attributes: *node*, *groups*, *status*, *statustime*, *appstatus*, *appstatustime*, *primarysn*, *hidden*, *updatestatus*, *updatestatustime*, *zonename*, *comments*, *disable*

DESCRIPTION

The list of all the nodes in the cluster, including each node's current status and what groups it is in.

nodelist Attributes:

node

> The hostname of a node in the cluster.

groups

> A comma-delimited list of groups this node is a member of. Group names are arbitrary, except all nodes should be part of the 'all' group. Internal group names are designated by using __<groupname>. For example, __Unmanaged, could be the internal name for a group of nodes that is not managed by xCAT. Admins should avoid using the __ characters when defining their groups.

status

> The current status of this node. This attribute will be set by xCAT software. Valid values: defined, booting, netbooting, booted, discovering, configuring, installing, alive, standingby, powering-off, unreachable. If blank, defined is assumed. The possible status change sequenses are: For installaton: defined->[discovering]->[configuring]->[standingby]->installing->booting->booted->[alive], For diskless deployment: defined->[discovering]->[configuring]->[standingby]->netbooting->booted->[alive], For booting: [alive/unreachable]->booting->[alive], For powering off: [alive]->powering-off->[unreachable], For monitoring: alive->unreachable. Discovering and configuring are for x Series dicovery process. Alive and unreachable are set only when there is a monitoring plug-in start monitor the node status for xCAT. Please note that the status values will not reflect the real node status if you change the state of the node from outside of xCAT (i.e. power off the node using HMC GUI).

statustime

> The data and time when the status was updated.

appstatus

> A comma-delimited list of application status. For example: 'sshd=up,ftp=down,ll=down'

appstatustime

> The date and time when appstatus was updated.

primarysn

> Not used currently. The primary servicenode, used by this node.

hidden

> Used to hide fsp and bpa definitions, 1 means not show them when running lsdef and nodels

updatestatus

> The current node update status. Valid states are synced, out-of-sync,syncing,failed.

updatestatustime

> The date and time when the updatestatus was updated.

zonename

> The name of the zone to which the node is currently assigned. If undefined, then it is not assigned to any zone.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

nodepos.5

NAME

nodepos - a table in the xCAT database.

SYNOPSIS

nodepos Attributes: *node*, *rack*, *u*, *chassis*, *slot*, *room*, *height*, *comments*, *disable*

DESCRIPTION

Contains info about the physical location of each node. Currently, this info is not used by xCAT, and therefore can be in whatevery format you want. It will likely be used in xCAT in the future.

nodepos Attributes:

node

> The node name or group name.

rack

> The frame the node is in.

u

> The vertical position of the node in the frame

chassis

> The BladeCenter chassis the blade is in.

slot

> The slot number of the blade in the chassis. For PCM, a comma-separated list of slot numbers is stored

room

> The room where the node is located.

height

> The server height in U(s).

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

noderes.5

NAME

noderes - a table in the xCAT database.

SYNOPSIS

noderes Attributes: *node*, *servicenode*, *netboot*, *tftpserver*, *tftpdir*, *nfsserver*, *monserver*, *nfsdir*, *installnic*, *primarynic*, *discoverynics*, *cmdinterface*, *xcatmaster*, *current_osimage*, *next_osimage*, *nimserver*, *routenames*, *nameservers*, *proxydhcp*, *comments*, *disable*

DESCRIPTION

Resources and settings to use when installing nodes.

noderes Attributes:

node

> The node name or group name.

servicenode

> A comma separated list of node names (as known by the management node) that provides most services for this node. The first service node on the list that is accessible will be used. The 2nd node on the list is generally considered to be the backup service node for this node when running commands like snmove.

netboot

> The type of network booting to use for this node. Valid values: pxe or xnba for x86* architecture, yaboot for POWER architecture, grub2-tftp and grub2-http for RHEL7 on Power and all the os deployment on Power LE. Notice: yaboot is not supported from rhels7 on Power,use grub2-tftp or grub2-http instead, the difference between the 2 is the file transfer protocol(i.e, http or tftp)

tftpserver

> The TFTP server for this node (as known by this node). If not set, it defaults to networks.tftpserver.

tftpdir

> The directory that roots this nodes contents from a tftp and related perspective. Used for NAS offload by using different mountpoints.

nfsserver

> The NFS or HTTP server for this node (as known by this node).

monserver

> The monitoring aggregation point for this node. The format is "x,y" where x is the ip address as known by the management node and y is the ip address as known by the node.

nfsdir

    The path that should be mounted from the NFS server.

installnic

    The network adapter on the node that will be used for OS deployment, the installnic can be set to the network adapter name or the mac address or the keyword "mac" which means that the network interface specified by the mac address in the mac table will be used. If not set, primarynic will be used. If primarynic is not set too, the keyword "mac" will be used as default.

primarynic

    This attribute will be deprecated. All the used network interface will be determined by installnic. The network adapter on the node that will be used for xCAT management, the primarynic can be set to the network adapter name or the mac address or the keyword "mac" which means that the network interface specified by the mac address in the mac table will be used. Default is eth0.

discoverynics

    If specified, force discovery to occur on specific network adapters only, regardless of detected connectivity. Syntax can be simply "eth2,eth3" to restrict discovery to whatever happens to come up as eth2 and eth3, or by driver name such as "bnx2:0,bnx2:1" to specify the first two adapters managed by the bnx2 driver

cmdinterface

    Not currently used.

xcatmaster

    The hostname of the xCAT service node (as known by this node). This acts as the default value for nfsserver and tftpserver, if they are not set. If xcatmaster is not set, the node will use whoever responds to its boot request as its master. For the directed bootp case for POWER, it will use the management node if xcatmaster is not set.

current_osimage

    Not currently used. The name of the osimage data object that represents the OS image currently deployed on this node.

next_osimage

    Not currently used. The name of the osimage data object that represents the OS image that will be installed on the node the next time it is deployed.

nimserver

    Not used for now. The NIM server for this node (as known by this node).

routenames

    A comma separated list of route names that refer to rows in the routes table. These are the routes that should be defined on this node when it is deployed.

nameservers

    An optional node/group specific override for name server list. Most people want to stick to site or network defined nameserver configuration.

proxydhcp

    To specify whether the node supports proxydhcp protocol. Valid values: yes or 1, no or 0. Default value is yes.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

nodetype.5

NAME

nodetype - a table in the xCAT database.

SYNOPSIS

nodetype Attributes: *node*, *os*, *arch*, *profile*, *provmethod*, *supportedarchs*, *nodetype*, *comments*, *disable*

DESCRIPTION

A few hardware and software characteristics of the nodes.

nodetype Attributes:

node

The node name or group name.

os

The operating system deployed on this node. Valid values: AIX, rhels*,rhelc*, rhas*,centos*,SL*, fedora*, sles* (where * is the version #). As a special case, if this is set to "boottarget", then it will use the initrd/kernel/parameters specified in the row in the boottarget table in which boottarget.bprofile equals nodetype.profile.

arch

The hardware architecture of this node. Valid values: x86_64, ppc64, x86, ia64.

profile

The string to use to locate a kickstart or autoyast template to use for OS deployment of this node. If the provmethod attribute is set to an osimage name, that takes precedence, and profile need not be defined. Otherwise, the os, profile, and arch are used to search for the files in /install/custom first, and then in /opt/xcat/share/xcat.

provmethod

The provisioning method for node deployment. The valid values are install, netboot, statelite or an os image name from the osimage table. If an image name is specified, the osimage definition stored in the osimage table and the linuximage table (for Linux) or nimimage table (for AIX) are used to locate the files

for templates, pkglists, syncfiles, etc. On Linux, if install, netboot or statelite is specified, the os, profile, and arch are used to search for the files in /install/custom first, and then in /opt/xcat/share/xcat.

supportedarchs

Comma delimited list of architectures this node can execute.

nodetype

A comma-delimited list of characteristics of this node. Valid values: ppc, blade, vm (virtual machine), osi (OS image), mm, mn, rsa, switch.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

notification.5

NAME

notification - a table in the xCAT database.

SYNOPSIS

notification Attributes: *filename*, *tables*, *tableops*, *comments*, *disable*

DESCRIPTION

Contains registrations to be notified when a table in the xCAT database changes. Users can add entries to have additional software notified of changes. Add and remove entries using the provided xCAT commands regnotif and unregnotif.

notification Attributes:

filename

The path name of a file that implements the callback routine when the monitored table changes. Can be a perl module or a command. See the regnotif man page for details.

tables

Comma-separated list of xCAT database tables to monitor.

tableops

Specifies the table operation to monitor for. Valid values: "d" (rows deleted), "a" (rows added), "u" (rows updated).

comments

>   Any user-written notes.

disable

>   Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

osdistro.5

NAME

osdistro - a table in the xCAT database.

SYNOPSIS

osdistro Attributes: *osdistroname*, *basename*, *majorversion*, *minorversion*, *arch*, *type*, *dirpaths*, *comments*, *disable*

DESCRIPTION

Information about all the OS distros in the xCAT cluster

osdistro Attributes:

osdistroname

>   Unique name (e.g. rhels6.2-x86_64)

basename

>   The OS base name (e.g. rhels)

majorversion

>   The OS distro major version.(e.g. 6)

minorversion

>   The OS distro minor version. (e.g. 2)

arch

>   The OS distro arch (e.g. x86_64)

type

>   Linux or AIX

dirpaths

>   Directory paths where OS distro is store. There could be multiple paths if OS distro has more than one
>   ISO image. (e.g. /install/rhels6.2/x86_64,...)

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

osdistroupdate.5

NAME

osdistroupdate - a table in the xCAT database.

SYNOPSIS

osdistroupdate Attributes: *osupdatename*, *osdistroname*, *dirpath*, *downloadtime*, *comments*, *disable*

DESCRIPTION

Information about the OS distro updates in the xCAT cluster

osdistroupdate Attributes:

osupdatename

> Name of OS update. (e.g. rhn-update1)

osdistroname

> The OS distro name to update. (e.g. rhels)

dirpath

> Path to where OS distro update is stored. (e.g. /install/osdistroupdates/rhels6.2-x86_64-20120716-update)

downloadtime

> The timestamp when OS distro update was downloaded..

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

osimage.5

NAME

osimage - a table in the xCAT database.

SYNOPSIS

osimage Attributes: *imagename*, *groups*, *profile*, *imagetype*, *description*, *provmethod*, *rootfstype*, *osdistroname*, *osupdatename*, *cfmdir*, *osname*, *osvers*, *osarch*, *synclists*, *postscripts*, *postbootscripts*, *serverrole*, *isdeletable*, *kitcomponents*, *comments*, *disable*

DESCRIPTION

Basic information about an operating system image that can be used to deploy cluster nodes.

osimage Attributes:

imagename

> The name of this xCAT OS image definition.

groups

> A comma-delimited list of image groups of which this image is a member. Image groups can be used in the litefile and litetree table instead of a single image name. Group names are arbitrary.

profile

> The node usage category. For example compute, service.

imagetype

> The type of operating system image this definition represents (linux,AIX).

description

> OS Image Description

provmethod

> The provisioning method for node deployment. The valid values are install, netboot,statelite,boottarget,dualboot,sysclone. If boottarget is set, you must set linuximage.boottarget to the name of the boottarget definition. It is not used by AIX.

rootfstype

> The filesystem type for the rootfs is used when the provmethod is statelite. The valid values are nfs or ramdisk. The default value is nfs

osdistroname

> The name of the OS distro definition. This attribute can be used to specify which OS distro to use, instead of using the osname,osvers,and osarch attributes. For *kit commands, the attribute will be used to read the osdistro table for the osname, osvers, and osarch attributes. If defined, the osname, osvers, and osarch attributes defined in the osimage table will be ignored.

osupdatename

A comma-separated list of OS distro updates to apply to this osimage.

cfmdir

CFM directory name for PCM. Set to /install/osimages/<osimage name>/cfmdir by PCM.

osname

Operating system name- AIX or Linux.

osvers

The Linux operating system deployed on this node. Valid values: rhels*,rhelc*, rhas*,centos*,SL*, fedora*, sles* (where * is the version #).

osarch

The hardware architecture of this node. Valid values: x86_64, ppc64, x86, ia64.

synclists

The fully qualified name of a file containing a list of files to synchronize on the nodes. Can be a comma separated list of multiple synclist files. The synclist generated by PCM named /install/osimages/<imagename>/synclist.cfm is reserved for use only by PCM and should not be edited by the admin.

postscripts

Comma separated list of scripts that should be run on this image after diskfull installation 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 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 installation of AIX, the scripts will run after the reboot and acts the same as the postbootscripts attribute. For AIX, use the postbootscripts attribute. See the site table runbootscripts attribute. Support will be added in the future for the postscripts attribute to run the scripts before the reboot in AIX.

postbootscripts

Comma separated list of scripts that should be run on this after diskfull installation or diskless boot. On AIX these scripts are run during the processing of /etc/inittab. On Linux they are run at the init.d time. xCAT automatically adds the scripts in the xcatdefaults.postbootscripts attribute to run first in the list. See the site table runbootscripts attribute.

serverrole

The role of the server created by this osimage. Default roles: mgtnode, servicenode, compute, login, storage, utility.

isdeletable

A flag to indicate whether this image profile can be deleted. This attribute is only used by PCM.

kitcomponents

List of Kit Component IDs assigned to this OS Image definition.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

passwd.5

NAME

passwd - a table in the xCAT database.

SYNOPSIS

passwd Attributes: *key*, *username*, *password*, *cryptmethod*, *authdomain*, *comments*, *disable*

DESCRIPTION

Contains default userids and passwords for xCAT to access cluster components. In most cases, xCAT will also actually set the userid/password in the relevant component when it is being configured or installed. Userids/passwords for specific cluster components can be overidden in other tables, e.g. mpa, ipmi, ppchcp, etc.

passwd Attributes:

key

> The type of component this user/pw is for. Valid values: blade (management module), ipmi (BMC), system (nodes), omapi (DHCP), hmc, ivm, cec, frame, switch.

username

> The default userid for this type of component

password

> The default password for this type of component

cryptmethod

> Indicates the method that was used to encrypt the password attribute. On AIX systems, if a value is provided for this attribute it indicates that the password attribute is encrypted. If the cryptmethod value is not set it indicates the password is a simple string value. On Linux systems, the cryptmethod is not supported however the code attempts to auto-discover MD5 encrypted passwords.

authdomain

> The domain in which this entry has meaning, e.g. specifying different domain administrators per active directory domain

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

performance.5

NAME

performance - a table in the xCAT database.

SYNOPSIS

performance Attributes: *timestamp*, *node*, *attrname*, *attrvalue*, *comments*, *disable*

DESCRIPTION

Describes the system performance every interval unit of time.

performance Attributes:

timestamp

> The time at which the metric was captured.

node

> The node name.

attrname

> The metric name.

attrvalue

> The metric value.

comments

> Any user-provided notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

policy.5

NAME

policy - a table in the xCAT database.

SYNOPSIS

policy Attributes: *priority*, *name*, *host*, *commands*, *noderange*, *parameters*, *time*, *rule*, *comments*, *disable*

DESCRIPTION

The policy table in the xCAT database controls who has authority to run specific xCAT operations. It is basically the Access Control List (ACL) for xCAT. It is sorted on the priority field before evaluating.

policy Attributes:

priority

> The priority value for this rule. This value is used to identify this policy data object (i.e. this rule) The table is sorted on this field with the lower the number the higher the priority. For example 1.0 is higher priority than 4.1 is higher than 4.9.

name

> The username that is allowed to perform the commands specified by this rule. Default is "*" (all users).

host

> The host from which users may issue the commands specified by this rule. Default is "*" (all hosts). Only all or one host is supported

commands

> The list of commands that this rule applies to. Default is "*" (all commands).

noderange

> The Noderange that this rule applies to. Default is "*" (all nodes). Not supported with the *def commands.

parameters

> A regular expression that matches the command parameters (everything except the noderange) that this rule applies to. Default is "*" (all parameters). Not supported with the *def commands.

time

> Time ranges that this command may be executed in. This is not supported.

rule

> Specifies how this rule should be applied. Valid values are: allow, accept, trusted. Allow or accept will allow the user to run the commands. Any other value will deny the user access to the commands. Trusted means that once this client has been authenticated via the certificate, all other information that is sent (e.g. the username) is believed without question. This authorization should only be given to the xcatd on the management node at this time.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

postscripts.5

NAME

postscripts - a table in the xCAT database.

SYNOPSIS

postscripts Attributes: *node*, *postscripts*, *postbootscripts*, *comments*, *disable*

DESCRIPTION

The scripts that should be run on each node after installation or diskless boot.

postscripts Attributes:

node

> The node name or group name.

postscripts

> Comma separated list of scripts that should be run on this node after diskfull installation or diskless boot. Each script can take zero or more parameters. For example: "script1 p1 p2,script2,...". 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 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 installation of AIX, the scripts will run after the reboot and acts the same as the postbootscripts attribute. For AIX, use the postbootscripts attribute. Support will be added in the future for the postscripts attribute to run the scripts before the reboot in AIX.

postbootscripts

> Comma separated list of scripts that should be run on this node after diskfull installation or diskless boot. Each script can take zero or more parameters. For example: "script1 p1 p2,script2,...". On AIX these scripts are run during the processing of /etc/inittab. On Linux they are run at the init.d time. xCAT automatically adds the scripts in the xcatdefaults.postbootscripts attribute to run first in the list.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

ppc.5

NAME

ppc - a table in the xCAT database.

SYNOPSIS

ppc Attributes: *node*, *hcp*, *id*, *pprofile*, *parent*, *nodetype*, *supernode*, *sfp*, *comments*, *disable*

DESCRIPTION

List of system p hardware: HMCs, IVMs, FSPs, BPCs, CECs, Frames.

ppc Attributes:

node

> The node name or group name.

hcp

> The hardware control point for this node (HMC, IVM, Frame or CEC). Do not need to set for BPAs and FSPs.

id

> For LPARs: the LPAR numeric id; for CECs: the cage number; for Frames: the frame number.

pprofile

> The LPAR profile that will be used the next time the LPAR is powered on with rpower. For DFM, the pprofile attribute should be set to blank

parent

> For LPARs: the CEC; for FSPs: the CEC; for CEC: the frame (if one exists); for BPA: the frame; for frame: the building block number (which consists 1 or more service nodes and compute/storage nodes that are serviced by them - optional).

nodetype

> The hardware type of the node. Only can be one of fsp, bpa, cec, frame, ivm, hmc and lpar

supernode

> Indicates the connectivity of this CEC in the HFI network. A comma separated list of 2 ids. The first one is the supernode number the CEC is part of. The second one is the logical location number (0-3) of this CEC within the supernode.

sfp

The Service Focal Point of this Frame. This is the name of the HMC that is responsible for collecting hardware service events for this frame and all of the CECs within this frame.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

ppcdirect.5

NAME

ppcdirect - a table in the xCAT database.

SYNOPSIS

ppcdirect Attributes: *hcp*, *username*, *password*, *comments*, *disable*

DESCRIPTION

Info necessary to use FSPs/BPAs to control system p CECs/Frames.

ppcdirect Attributes:

hcp

Hostname of the FSPs/BPAs(for ASMI) and CECs/Frames(for DFM).

username

Userid of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

password

Password of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

ppchcp.5

NAME

ppchcp - a table in the xCAT database.

SYNOPSIS

ppchcp Attributes: *hcp*, *username*, *password*, *comments*, *disable*

DESCRIPTION

Info necessary to use HMCs and IVMs as hardware control points for LPARs.

ppchcp Attributes:

hcp

> Hostname of the HMC or IVM.

username

> Userid of the HMC or IVM. If not filled in, xCAT will look in the passwd table for key=hmc or key=ivm.
> If not in the passwd table, the default used is hscroot for HMCs and padmin for IVMs.

password

> Password of the HMC or IVM. If not filled in, xCAT will look in the passwd table for key=hmc or
> key=ivm. If not in the passwd table, the default used is abc123 for HMCs and padmin for IVMs.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

prescripts.5

NAME

prescripts - a table in the xCAT database.

SYNOPSIS

prescripts Attributes: *node*, *begin*, *end*, *comments*, *disable*

DESCRIPTION

The scripts that will be run at the beginning and the end of the nodeset(Linux), nimnodeset(AIX) or mkdsklsnode(AIX) command.

prescripts Attributes:

node

> The node name or group name.

begin

> **The scripts to be run at the beginning of the nodeset(Linux),** nimnodeset(AIX) or mkdskl-
> snode(AIX) command. The format is:
>
> > [action1:]s1,s2...[|action2:s3,s4,s5...]
>
> **where:**
>
> > - action1 and action2 for Linux are the nodeset actions specified in the command. For AIX, action1 and action1 can be 'diskless' for mkdsklsnode command' and 'standalone for nimnodeset command.
> > - 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 (all actions) diskless:myscript1,myscript2 (AIX) in-
> > stall:myscript1,myscript2|netboot:myscript3 (Linux)

```
All the scripts should be copied to /install/prescripts directory.
The following two environment variables will be passed to each script:
  NODES a coma separated list of node names that need to run the script for
  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
will be invoked at at a time. If it is not specified, the script will be␣
↪invoked
once for all the nodes.
```

end

> **The scripts to be run at the end of the nodeset(Linux),** nimnodeset(AIX),or mkdsklsnode(AIX) com-
> mand. The format is the same as the 'begin' column.

comments

> Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

prodkey.5

NAME

prodkey - a table in the xCAT database.

SYNOPSIS

prodkey Attributes: *node*, *product*, *key*, *comments*, *disable*

DESCRIPTION

Specify product keys for products that require them

prodkey Attributes:

node

> The node name or group name.

product

> A string to identify the product (for OSes, the osname would be used, i.e. wink28

key

> The product key relevant to the aforementioned node/group and product combination

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

rack.5

NAME

rack - a table in the xCAT database.

SYNOPSIS

rack Attributes: *rackname*, *displayname*, *num*, *height*, *room*, *comments*, *disable*

DESCRIPTION

Rack information.

rack Attributes:

rackname

> The rack name.

displayname

> Alternative name for rack. Only used by PCM.

num

> The rack number.

height

> Number of units which can be stored in the rack.

room

> The room in which the rack is located.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

routes.5

NAME

routes - a table in the xCAT database.

SYNOPSIS

routes Attributes: *routename*, *net*, *mask*, *gateway*, *ifname*, *comments*, *disable*

DESCRIPTION

Describes the additional routes needed to be setup in the os routing table. These routes usually are used to connect the management node to the compute node using the servie node as gateway.

routes Attributes:

routename

>   Name used to identify this route.

net

>   The network address.

mask

>   The network mask.

gateway

>   The gateway that routes the ip traffic from the mn to the nodes. It is usually a service node.

ifname

>   The interface name that facing the gateway. It is optional for IPv4 routes, but it is required for IPv6 routes.

comments

>   Any user-written notes.

disable

>   Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

servicenode.5

NAME

servicenode - a table in the xCAT database.

SYNOPSIS

servicenode Attributes: *node*, *nameserver*, *dhcpserver*, *tftpserver*, *nfsserver*, *conserver*, *monserver*, *ldapserver*, *ntpserver*, *ftpserver*, *nimserver*, *ipforward*, *dhcpinterfaces*, *proxydhcp*, *comments*, *disable*

DESCRIPTION

List of all Service Nodes and services that will be set up on the Service Node.

servicenode Attributes:

node

> The hostname of the service node as known by the Management Node.

nameserver

> Do we set up DNS on this service node? Valid values: 2, 1, no or 0. If 2, creates named.conf as dns slave, using the management node as dns master, and starts named. If 1, creates named.conf file with forwarding to the management node and starts named. If no or 0, it does not change the current state of the service.

dhcpserver

> Do we set up DHCP on this service node? Not supported on AIX. Valid values:yes or 1, no or 0. If yes, runs makedhcp -n. If no or 0, it does not change the current state of the service.

tftpserver

> Do we set up TFTP on this service node? Not supported on AIX. Valid values:yes or 1, no or 0. If yes, configures and starts atftp. If no or 0, it does not change the current state of the service.

nfsserver

> Do we set up file services (HTTP,FTP,or NFS) on this service node? For AIX will only setup NFS, not HTTP or FTP. Valid values:yes or 1, no or 0.If no or 0, it does not change the current state of the service.

conserver

> Do we set up Conserver on this service node? Valid values:yes or 1, no or 0. If yes, configures and starts conserver daemon. If no or 0, it does not change the current state of the service.

monserver

> Is this a monitoring event collection point? Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

ldapserver

> Do we set up ldap caching proxy on this service node? Not supported on AIX. Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

ntpserver

> Not used. Use setupntp postscript to setup a ntp server on this service node? Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

ftpserver

> Do we set up a ftp server on this service node? Not supported on AIX Valid values:yes or 1, no or 0. If yes, configure and start vsftpd. (You must manually install vsftpd on the service nodes before this.) If no or 0, it does not change the current state of the service. xCAT is not using ftp for compute nodes provisioning or any other xCAT features, so this attribute can be set to 0 if the ftp service will not be used for other purposes

nimserver

> Not used. Do we set up a NIM server on this service node? Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

ipforward

> Do we set up ip forwarding on this service node? Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

dhcpinterfaces

The network interfaces DHCP server should listen on for the target node. This attribute can be used for management node and service nodes. If defined, it will override the values defined in site.dhcpinterfaces. This is a comma separated list of device names. !remote! indicates a non-local network for relay DHCP. For example: !remote!,eth0,eth1

proxydhcp

Do we set up proxydhcp service on this node? valid values: yes or 1, no or 0. If yes, the proxydhcp daemon will be enabled on this node.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

site.5

NAME

site - a table in the xCAT database.

SYNOPSIS

site Attributes: *key*, *value*, *comments*, *disable*

DESCRIPTION

Global settings for the whole cluster. This table is different from the other tables in that each attribute is just named in the key column, rather than having a separate column for each attribute. The following is a list of attributes currently used by xCAT organized into categories.

site Attributes:

key

Attribute Name: Description

```
 ------------
AIX ATTRIBUTES
 ------------
 nimprime :   The name of NIM server, if not set default is the AIX MN.
             If Linux MN, then must be set for support of mixed cluster␣
→(TBD).

 useSSHonAIX:  (yes/1 or no/0). Default is yes.  The support for rsh/rcp is␣
→deprecated.
```

```
 useNFSv4onAIX:  (yes/1 or no/0). If yes, NFSv4 will be used with NIM. If no,
                 NFSv3 will be used with NIM. Default is no.


 -----------------
DATABASE ATTRIBUTES
 -----------------
 auditnosyslog: If set to 1, then commands will only be written to the␣
 ↪auditlog table.
                 This attribute set to 1 and auditskipcmds=ALL means no␣
 ↪logging of commands.
                 Default is to write to both the auditlog table and syslog.
 auditskipcmds: List of commands and/or client types that will not be
                 written to the auditlog table and syslog. See auditnosyslog.
                 'ALL' means all cmds will be skipped. If attribute is null,␣
 ↪all
                 commands will be written.
                 clienttype:web would skip all commands from the web client
                 For example: tabdump,nodels,clienttype:web
                 will not log tabdump,nodels and any web client commands.

 databaseloc:    Directory where we create the db instance directory.
                  Default is /var/lib. Only DB2 is currently supported.
                  Do not use the directory in the site.installloc or
                  installdir attribute. This attribute must not be changed
                  once db2sqlsetup script has been run and DB2 has been setup.

 excludenodes:  A set of comma separated nodes and/or groups that would␣
 ↪automatically
                 be subtracted from any noderange, it can be used for␣
 ↪excluding some
                 failed nodes for any xCAT commands. See the 'noderange'␣
 ↪manpage for
                 details on supported formats.

 nodestatus:  If set to 'n', the nodelist.status column will not be updated␣
 ↪during
                 the node deployment, node discovery and power operations. The␣
 ↪default is to update.

 skiptables:  Comma separated list of tables to be skipped by dumpxCATdb

 skipvalidatelog:  If set to 1, then getcredentials and getpostscripts calls␣
 ↪will not be logged in syslog.

 -------------
DHCP ATTRIBUTES
 -------------
 dhcpinterfaces:  The network interfaces DHCP should listen on.  If it is␣
 ↪the same
                    for all nodes, use a simple comma-separated list of NICs. ␣
 ↪To
                    specify different NICs for different nodes:
                       xcatmn|eth1,eth2;service|bond0.
                    In this example xcatmn is the name of the xCAT MN, and␣
 ↪DHCP there
                    should listen on eth1 and eth2.  On all of the nodes in␣
 ↪group
                    'service' DHCP should listen on the bond0 nic.
```

```
 dhcpsetup:  If set to 'n', it will skip the dhcp setup process in the␣
↪nodeset cmd.

 dhcplease:  The lease time for the dhcp client. The default value is 43200.

 disjointdhcps:  If set to '1', the .leases file on a service node only␣
↪contains
                the nodes it manages. The default value is '0'.
                '0' value means include all the nodes in the subnet.

 pruneservices:  Whether to enable service pruning when noderm is run (i.e.
                removing DHCP entries when noderm is executed)

 managedaddressmode: The mode of networking configuration during node␣
↪provision.
                    If set to 'static', the network configuration will be␣
↪configured
                    in static mode based on the node and network definition␣
↪on MN.
                    If set to 'dhcp', the network will be configured with␣
↪dhcp protocol.
                    The default is 'dhcp'.

 ------------
DNS ATTRIBUTES
 ------------
 dnshandler:  Name of plugin that handles DNS setup for makedns.
 domain:  The DNS domain name used for the cluster.

 forwarders:  The DNS servers at your site that can provide names outside of␣
↪the
              cluster. The makedns command will configure the DNS on the␣
↪management
              node to forward requests it does not know to these servers.
              Note that the DNS servers on the service nodes will ignore␣
↪this value
              and always be configured to forward requests to the management␣
↪node.

 master:  The hostname of the xCAT management node, as known by the nodes.

 nameservers:  A comma delimited list of DNS servers that each node in the␣
↪cluster
               should use. This value will end up in the nameserver settings␣
↪of the
               /etc/resolv.conf on each node. It is common (but not␣
↪required) to set
               this attribute value to the IP addr of the xCAT management␣
↪node, if
               you have set up the DNS on the management node by running␣
↪makedns.
               In a hierarchical cluster, you can also set this attribute to
               "<xcatmaster>" to mean the DNS server for each node should be␣
↪the
               node that is managing it (either its service node or the␣
↪management
               node).
```

```
externaldns:  To specify that external dns is used. If externaldns is set␣
→to any value
             then, makedns command will not start the local nameserver on␣
→xCAT MN.
             Default is to start the local nameserver.

dnsupdaters:  The value are ',' separated string which will be added to the␣
→zone config
             section. This is an interface for user to add configuration␣
→entries to
             the zone sections in named.conf.

dnsinterfaces:  The network interfaces DNS server should listen on.  If it␣
→is the same
               for all nodes, use a simple comma-separated list of NICs.  ␣
→To
               specify different NICs for different nodes:
                   xcatmn|eth1,eth2;service|bond0.
               In this example xcatmn is the name of the xCAT MN, and DNS␣
→there
               should listen on eth1 and eth2.  On all of the nodes in␣
→group
               'service' DNS should listen on the bond0 nic.
               NOTE: if using this attribute to block certain interfaces,␣
→make sure
               the ip maps to your hostname of xCAT MN is not blocked␣
→since xCAT needs to
               use this ip to communicate with the local NDS server on MN.

 ------------------------
HARDWARE CONTROL ATTRIBUTES
 ------------------------
blademaxp:  The maximum number of concurrent processes for blade hardware␣
→control.

ea_primary_hmc:  The hostname of the HMC that the Integrated Switch Network
                Management Event Analysis should send hardware serviceable
                events to for processing and potentially sending to IBM.

ea_backup_hmc:  The hostname of the HMC that the Integrated Switch Network
               Management Event Analysis should send hardware serviceable
               events to if the primary HMC is down.

enableASMI:  (yes/1 or no/0). If yes, ASMI method will be used after fsp-
→api. If no,
            when fsp-api is used, ASMI method will not be used. Default␣
→is no.

fsptimeout:  The timeout, in milliseconds, to use when communicating with␣
→FSPs.

hwctrldispatch:  Whether or not to send hw control operations to the service
                node of the target nodes. Default is 'y'.(At present, this␣
→attribute
                is only used for IBM Flex System)

ipmidispatch:  Whether or not to send ipmi hw control operations to the␣
→service
```

```
                    node of the target compute nodes. Default is 'y'.

 ipmimaxp:  The max # of processes for ipmi hw ctrl. The default is 64.␣
 →Currently,
            this is only used for HP hw control.

 ipmiretries:  The # of retries to use when communicating with BMCs. Default␣
 →is 3.

 ipmisdrcache:  If set to 'no', then the xCAT IPMI support will not cache␣
 →locally
                the target node's SDR cache to improve performance.

 ipmitimeout:  The timeout to use when communicating with BMCs. Default is 2.
               This attribute is currently not used.

 maxssh:  The max # of SSH connections at any one time to the hw ctrl point␣
 →for PPC
          This parameter doesn't take effect on the rpower command.
          It takes effects on other PPC hardware control command
          getmacs/rnetboot/rbootseq and so on. Default is 8.

 syspowerinterval:  For system p CECs, this is the number of seconds the␣
 →rpower
                    command will wait between performing the action for each␣
 →CEC.
                    For system x IPMI servers, this is the number of seconds the
                    rpower command will wait between powering on
 →<syspowermaxnodes>
                    nodes at a time.  This value is used to control the power␣
 →on speed
                    in large clusters. Default is 0.

 syspowermaxnodes:  The number of servers to power on at one time before␣
 →waiting
                    'syspowerinterval' seconds to continue on to the next␣
 →set of
                    nodes.  If the noderange given to rpower includes nodes␣
 →served
                    by different service nodes, it will try to spread each␣
 →set of
                    nodes across the service nodes evenly. Currently only␣
 →used for
                    IPMI servers and must be set if 'syspowerinterval' is␣
 →set.

 powerinterval:  The number of seconds the rpower command to LPARs will wait␣
 →between
                 performing the action for each LPAR. LPARs of different HCPs
                 (HMCs or FSPs) are done in parallel. This is used to limit␣
 →the
                 cluster boot up speed in large clusters. Default is 0. ␣
 →This is
                 currently only used for system p hardware.

 ppcmaxp:  The max # of processes for PPC hw ctrl. If there are more than␣
 →ppcmaxp
           hcps, this parameter will take effect. It will control the max␣
 →number of
```

```
          processes for PPC hardware control commands. Default is 64.

 ppcretry:  The max # of PPC hw connection attempts to HMC before failing.
            It only takes effect on the hardware control commands through HMC.
            Default is 3.

 ppctimeout:  The timeout, in milliseconds, to use when communicating with␣
→PPC hw
              through HMC. It only takes effect on the hardware control␣
→commands
              through HMC. Default is 0.

 snmpc:  The snmp community string that xcat should use when communicating␣
→with the
         switches.

 --------------------------
 INSTALL/DEPLOYMENT ATTRIBUTES
 --------------------------
 cleanupxcatpost:  (yes/1 or no/0). Set to 'yes' or '1' to clean up the /
→xcatpost
                   directory on the stateless and statelite nodes after the
                   postscripts are run. Default is no.

 db2installloc:  The location which the service nodes should mount for
                 the db2 code to install. Format is hostname:/path.  If␣
→hostname is
                 omitted, it defaults to the management node. Default is /
→mntdb2.

 defserialflow:  The default serial flow – currently only used by the mknb␣
→command.

 defserialport:  The default serial port – currently only used by mknb.

 defserialspeed:  The default serial speed – currently only used by mknb.

 genmacprefix:  When generating mac addresses automatically, use this␣
→manufacturing
                prefix (e.g. 00:11:aa)

 genpasswords:  Automatically generate random passwords for BMCs when␣
→configuring
                them.

 installdir:  The local directory name used to hold the node deployment␣
→packages.

 installloc:  The location from which the service nodes should mount the
              deployment packages in the format hostname:/path.  If hostname␣
→is
              omitted, it defaults to the management node. The path must
              match the path in the installdir attribute.

 iscsidir:  The path to put the iscsi disks in on the mgmt node.

 mnroutenames:  The name of the routes to be setup on the management node.
                It is a comma separated list of route names that are defined␣
→in the
```

```
                     routes table.

 runbootscripts:   If set to 'yes' the scripts listed in the postbootscripts
                     attribute in the osimage and postscripts tables will be␣
→run during
                     each reboot of stateful (diskful) nodes. This attribute␣
→has no
                     effect on stateless and statelite nodes. Please run the␣
→following
                     command after you change the value of this attribute:
                     'updatenode <nodes> -P setuppostbootscripts'

 precreatemypostscripts: (yes/1 or no/0). Default is no. If yes, it will
               instruct xCAT at nodeset and updatenode time to query the db␣
→once for
               all of the nodes passed into the cmd and create the␣
→mypostscript file
               for each node, and put them in a directory of tftpdir(such as:␣
→/tftpboot)
               If no, it will not generate the mypostscript file in the␣
→tftpdir.

 setinstallnic:  Set the network configuration for installnic to be static.

 sharedtftp:  Set to 0 or no, xCAT should not assume the directory
               in tftpdir is mounted on all on Service Nodes. Default is 1/
→yes.
               If value is set to a hostname, the directory in tftpdir
               will be mounted from that hostname on the SN

 sharedinstall: Indicates if a shared file system will be used for␣
→installation
                 resources. Possible values are: 'no', 'sns', or 'all'.  'no'
                 means a shared file system is not being used.  'sns' means a
                 shared filesystem is being used across all service nodes.
                 'all' means that the management as well as the service nodes
                 are all using a common shared filesystem. The default is 'no'.
 xcatconfdir:  Where xCAT config data is (default /etc/xcat).

 --------------------
REMOTESHELL ATTRIBUTES
 --------------------
 nodesyncfiledir:  The directory on the node, where xdcp will rsync the files
 SNsyncfiledir:  The directory on the Service Node, where xdcp will rsync␣
→the files
                 from the MN that will eventually be rsync'd to the compute␣
→nodes.

 sshbetweennodes:  Comma separated list of groups of compute nodes to enable␣
→passwordless root
                     ssh during install, or xdsh -K. Default is ALLGROUPS.
                     Set to NOGROUPS,if you do not wish to enabled any group␣
→of compute nodes.
                     Service Nodes are not affected by this attribute
                     they are always setup with
                     passwordless root access to nodes and other SN.
                     If using the zone table, this attribute in not used.
```

```
-----------------
SERVICES ATTRIBUTES
-----------------
 consoleondemand:  When set to 'yes', conserver connects and creates the␣
→console
                   output only when the user opens the console. Default is␣
→no on
                   Linux, yes on AIX.

 consoleservice:   The console service to be used by xCAT. Default is␣
→conserver

 httpport:    The port number that the booting/installing nodes should␣
→contact the
              http server on the MN/SN on. It is your responsibility to␣
→configure
              the http server to listen on that port - xCAT will not do that.

 nmapoptions: Additional options for the nmap command. nmap is used in pping,
              nodestat, xdsh -v and updatenode commands. Sometimes additional
              performance tuning may be needed for nmap due to network␣
→traffic.
              For example, if the network response time is too slow, nmap␣
→may not
              give stable output. You can increase the timeout value by␣
→specifying
              '--min-rtt-timeout 1s'. xCAT will append the options defined␣
→here to
              the nmap command.

 ntpservers:  A comma delimited list of NTP servers for the cluster - often␣
→the
              xCAT management node.

 svloglocal:  if set to 1, syslog on the service node will not get forwarded␣
→to the
              mgmt node.

 timezone:  (e.g. America/New_York)

 tftpdir:  tftp directory path. Default is /tftpboot

 tftpflags:  The flags that used to start tftpd. Default is '-v -l -s /
→tftpboot
              -m /etc/tftpmapfile4xcat.conf' if tftplfags is not set

 useNmapfromMN:  When set to yes, nodestat command should obtain the node␣
→status
                 using nmap (if available) from the management node instead␣
→of the
                 service node. This will improve the performance in a flat␣
→network.

 vsftp:      Default is 'n'. If set to 'y', the xcatd on the mn will␣
→automatically
              bring up vsftpd.  (You must manually install vsftpd before␣
→this.
              This setting does not apply to the service node. For sn
```

```
                you need to set servicenode.ftpserver=1 if you want xcatd to
                bring up vsftpd.

 -----------------------
VIRTUALIZATION ATTRIBUTES
 -----------------------
 usexhrm:  Have xCAT run its xHRM script when booting up KVM guests to set␣
→the
            virtual network bridge up correctly. See
            https://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_
→Virtualization_with_KVM#Setting_up_a_network_bridge

 vcenterautojoin:  When set to no, the VMWare plugin will not attempt to␣
→auto remove
                    and add hypervisors while trying to perform operations. ␣
→If users
                    or tasks outside of xCAT perform the joining this assures␣
→xCAT
                    will not interfere.

 vmwarereconfigonpower:  When set to no, the VMWare plugin will make no␣
→effort to
                          push vm.cpus/vm.memory updates from xCAT to VMWare.

 persistkvmguests:  Keep the kvm definition on the kvm hypervisor when you␣
→power off
                      the kvm guest node. This is useful for you to manually␣
→change the
                      kvm xml definition file in virsh for debugging. Set␣
→anything means
                      enable.


 --------------------
XCAT DAEMON ATTRIBUTES
 --------------------
 useflowcontrol:  (yes/1 or no/0). If yes, the postscript processing on each␣
→node
                contacts xcatd on the MN/SN using a lightweight UDP packet to␣
→wait
                until xcatd is ready to handle the requests associated with
                postscripts.  This prevents deploying nodes from flooding␣
→xcatd and
                locking out admin interactive use. This value works with the
                xcatmaxconnections and xcatmaxbatch attributes. Is not␣
→supported on AIX.
                If the value is no, nodes sleep for a random time before␣
→contacting
                xcatd, and retry. The default is no.
                See the following document for details:
                Hints_and_Tips_for_Large_Scale_Clusters

 xcatmaxconnections:  Number of concurrent xCAT protocol requests before␣
→requests
                        begin queueing. This applies to both client command␣
→requests
                        and node requests, e.g. to get postscripts. Default is␣
→64.
```

```
 xcatmaxbatchconnections:  Number of concurrent xCAT connections allowed␣
↪from the nodes.
                          Value must be less than xcatmaxconnections. Default is␣
↪50.

 xcatdport:  The port used by the xcatd daemon for client/server␣
↪communication.

 xcatiport:  The port used by xcatd to receive install status updates from␣
↪nodes.

 xcatsslversion:  The ssl version by xcatd. Default is SSLv3.

 xcatsslciphers:  The ssl cipher by xcatd. Default is 3DES.
```

value

> The value of the attribute specified in the "key" column.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

statelite.5

NAME

statelite - a table in the xCAT database.

SYNOPSIS

statelite Attributes: *node*, *image*, *statemnt*, *mntopts*, *comments*, *disable*

DESCRIPTION

The location on an NFS server where a nodes persistent files are stored. Any file marked persistent in the litefile table will be stored in the location specified in this table for that node.

statelite Attributes:

node

> The name of the node or group that will use this location.

image

Reserved for future development, not used.

statemnt

The persistant read/write area where a node's persistent files will be written to, e.g: 10.0.0.1/state/. The node name will be automatically added to the pathname, so 10.0.0.1:/state, will become 10.0.0.1:/state/<nodename>.

mntopts

A comma-separated list of options to use when mounting the persistent directory. (Ex. 'soft') The default is to do a 'hard' mount.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

storage.5

NAME

storage - a table in the xCAT database.

SYNOPSIS

storage Attributes: *node*, *osvolume*, *size*, *state*, *storagepool*, *hypervisor*, *fcprange*, *volumetag*, *type*, *controller*, *comments*, *disable*

DESCRIPTION

storage Attributes:

node

The node name

osvolume

Specification of what storage to place the node OS image onto. Examples include: localdisk (Install to first non-FC attached disk) usbdisk (Install to first USB mass storage device seen) wwn=0x50000393c813840c (Install to storage device with given WWN)

size

Size of the volume. Examples include: 10G, 1024M.

state

State of the volume. The valid values are: free, used, and allocated

storagepool

> Name of storage pool where the volume is assigned.

hypervisor

> Name of the hypervisor where the volume is configured.

fcprange

> A range of acceptable fibre channels that the volume can use. Examples include: 3B00-3C00;4B00-4C00.

volumetag

> A specific tag used to identify the volume in the autoyast or kickstart template.

type

> The plugin used to drive storage configuration (e.g. svc)

controller

> **The management address to attach/detach new volumes.** In the scenario involving multiple controllers, this data must be passed as argument rather than by table value

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

switch.5

NAME

switch - a table in the xCAT database.

SYNOPSIS

switch Attributes: *node*, *switch*, *port*, *vlan*, *interface*, *comments*, *disable*

DESCRIPTION

Contains what switch port numbers each node is connected to.

switch Attributes:

node

> The node name or group name.

switch

> The switch hostname.

port

> The port number in the switch that this node is connected to. On a simple 1U switch, an administrator can generally enter the number as printed next to the ports, and xCAT will understand switch representation differences. On stacked switches or switches with line cards, administrators should usually use the CLI representation (i.e. 2/0/1 or 5/8). One notable exception is stacked SMC 8848M switches, in which you must add 56 for the proceeding switch, then the port number. For example, port 3 on the second switch in an SMC8848M stack would be 59

vlan

> The ID for the tagged vlan that is created on this port using mkvlan and chvlan commands.

interface

> The interface name from the node perspective. For example, eth0. For the primary nic, it can be empty, the word "primary" or "primary:ethx" where ethx is the interface name.

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

switches.5

NAME

switches - a table in the xCAT database.

SYNOPSIS

switches Attributes: *switch*, *snmpversion*, *username*, *password*, *privacy*, *auth*, *linkports*, *sshusername*, *sshpassword*, *protocol*, *switchtype*, *comments*, *disable*

DESCRIPTION

Parameters to use when interrogating switches

switches Attributes:

switch

> The hostname/address of the switch to which the settings apply

snmpversion

> The version to use to communicate with switch. SNMPv1 is assumed by default.

username

> The username to use for SNMPv3 communication, ignored for SNMPv1

password

> The password or community string to use for SNMPv3 or SNMPv1 respectively. Falls back to passwd table, and site snmpc value if using SNMPv1

privacy

> The privacy protocol to use for v3. DES is assumed if v3 enabled, as it is the most readily available.

auth

> The authentication protocol to use for SNMPv3. SHA is assumed if v3 enabled and this is unspecified

linkports

> The ports that connect to other switches. Currently, this column is only used by vlan configuration. The format is: "port_number:switch,port_number:switch...". Please refer to the switch table for details on how to specify the port numbers.

sshusername

> The remote login user name. It can be for ssh or telnet. If it is for telnet, please set protocol to "telnet".

sshpassword

> The remote login password. It can be for ssh or telnet. If it is for telnet, please set protocol to "telnet".

protocol

> Prorocol for running remote commands for the switch. The valid values are: ssh, telnet. ssh is the default. Leave it blank or set to "ssh" for Mellanox IB switch.

switchtype

> The type of switch. It is used to identify the file name that implements the functions for this swithc. The valid values are: MellanoxIB etc.

comments

disable

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

token.5

NAME

token - a table in the xCAT database.

SYNOPSIS

token Attributes: *tokenid*, *username*, *expire*, *comments*, *disable*

DESCRIPTION

The token of users for authentication.

token Attributes:

tokenid

It is a UUID as an unified identify for the user.

username

The user name.

expire

The expire time for this token.

comments

Any user-provided notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

virtsd.5

NAME

virtsd - a table in the xCAT database.

SYNOPSIS

virtsd Attributes: *node*, *sdtype*, *stype*, *location*, *host*, *cluster*, *datacenter*, *comments*, *disable*

DESCRIPTION

The parameters which used to create the Storage Domain

virtsd Attributes:

node

> The name of the storage domain

sdtype

> The type of storage domain. Valid values: data, iso, export

stype

> The type of storge. Valid values: nfs, fcp, iscsi, localfs

location

> The path of the storage

host

> For rhev, a hypervisor host needs to be specified to manage the storage domain as SPM (Storage Pool Manager). But the SPM role will be failed over to another host when this host down.

cluster

> A cluster of hosts

datacenter

> A collection for all host, vm that will shared the same storages, networks.

comments

disable

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

vm.5

NAME

vm - a table in the xCAT database.

SYNOPSIS

vm Attributes: *node*, *mgr*, *host*, *migrationdest*, *storage*, *storagemodel*, *storagecache*, *storageformat*, *cfgstore*, *memory*, *cpus*, *nics*, *nicmodel*, *bootorder*, *clockoffset*, *virtflags*, *master*, *vncport*, *textconsole*, *powerstate*, *beacon*, *datacenter*, *cluster*, *guestostype*, *othersettings*, *physlots*, *vidmodel*, *vidproto*, *vidpassword*, *comments*, *disable*

DESCRIPTION

Virtualization parameters

vm Attributes:

node

    The node or static group name

mgr

    The function manager for the virtual machine

host

    The system that currently hosts the VM

migrationdest

    A noderange representing candidate destinations for migration (i.e. similar systems, same SAN, or other criteria that xCAT can use

storage

    A list of storage files or devices to be used. i.e. dir:///cluster/vm/<nodename> or nfs://<server>/path/to/folder/

storagemodel

    Model of storage devices to provide to guest

storagecache

    Select caching scheme to employ. E.g. KVM understands 'none', 'writethrough' and 'writeback'

storageformat

    Select disk format to use by default (e.g. raw versus qcow2)

cfgstore

    Optional location for persistant storage separate of emulated hard drives for virtualization solutions that require persistant store to place configuration data

memory

    Megabytes of memory the VM currently should be set to.

cpus

    Number of CPUs the node should see.

nics

    Network configuration parameters. Of the general form [physnet:]interface,.. Generally, interface describes the vlan entity (default for native, tagged for tagged, vl[number] for a specific vlan. physnet is a virtual switch name or port description that is used for some virtualization technologies to construct virtual switches. hypervisor.netmap can map names to hypervisor specific layouts, or the descriptions described there may be used directly here where possible.

nicmodel

    Model of NICs that will be provided to VMs (i.e. e1000, rtl8139, virtio, etc)

bootorder

    Boot sequence (i.e. net,hd)

clockoffset

Whether to have guest RTC synced to "localtime" or "utc" If not populated, xCAT will guess based on the nodetype.os contents.

virtflags

General flags used by the virtualization method. For example, in Xen it could, among other things, specify paravirtualized

imageformat=[raw|fullraw|qcow2] raw is a generic sparse file that allocates storage on demand fullraw is a generic, non-sparse file that preallocates all space qcow2 is a sparse, copy-on-write capable format implemented at the virtualization layer rather than the filesystem level

clonemethod=[qemu-img|reflink] qemu-img allows use of qcow2 to generate virtualization layer copy-on-write reflink uses a generic filesystem facility to clone the files on your behalf, but requires filesystem support such as btrfs

placement_affinity=[migratable|user_migratable|pinned]

master

The name of a master image, if any, this virtual machine is linked to. This is generally set by clonevm and indicates the deletion of a master that would invalidate the storage of this virtual machine

vncport

Tracks the current VNC display port (currently not meant to be set

textconsole

Tracks the Psuedo-TTY that maps to the serial port or console of a VM

powerstate

This flag is used by xCAT to track the last known power state of the VM.

beacon

This flag is used by xCAT to track the state of the identify LED with respect to the VM.

datacenter

Optionally specify a datacenter for the VM to exist in (only applicable to VMWare)

cluster

Specify to the underlying virtualization infrastructure a cluster membership for the hypervisor.

guestostype

This allows administrator to specify an identifier for OS to pass through to virtualization stack. Normally this should be ignored as xCAT will translate from nodetype.os rather than requiring this field be used

othersettings

This allows specifying a semicolon delimited list of key->value pairs to include in a vmx file of VMware. For partitioning on normal power machines, this option is used to specify the hugepage and/or bsr information, the value is like:'hugepage:1,bsr=2'.

physlots

Specify the physical slots drc index that will assigned to the partition, the delimiter is ',', and the drc index must started with '0x'. For more details, please reference to manpage of 'lsvm'.

vidmodel

Model of video adapter to provide to guest. For example, qxl in KVM

vidproto

Request a specific protocol for remote video access be set up. For example, spice in KVM.

vidpassword

Password to use instead of temporary random tokens for VNC and SPICE access

comments

disable

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

vmmaster.5

NAME

vmmaster - a table in the xCAT database.

SYNOPSIS

vmmaster Attributes: *name*, *os*, *arch*, *profile*, *storage*, *storagemodel*, *nics*, *vintage*, *originator*, *virttype*, *specializeparameters*, *comments*, *disable*

DESCRIPTION

Inventory of virtualization images for use with clonevm. Manual intervention in this table is not intended.

vmmaster Attributes:

name

The name of a master

os

The value of nodetype.os at the time the master was captured

arch

The value of nodetype.arch at the time of capture

profile

The value of nodetype.profile at time of capture

storage

The storage location of bulk master information

storagemodel

The default storage style to use when modifying a vm cloned from this master

nics

The nic configuration and relationship to vlans/bonds/etc

vintage

When this image was created

originator

The user who created the image

virttype

The type of virtualization this image pertains to (e.g. vmware, kvm, etc)

specializeparameters

Implementation specific arguments, currently only "autoLogonCount=<number" for ESXi clonevme

comments

disable

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

vpd.5

NAME

vpd - a table in the xCAT database.

SYNOPSIS

vpd Attributes: *node*, *serial*, *mtm*, *side*, *asset*, *uuid*, *comments*, *disable*

DESCRIPTION

The Machine type, Model, and Serial numbers of each node.

vpd Attributes:

node

The node name or group name.

serial

The serial number of the node.

mtm

The machine type and model number of the node. E.g. 7984-6BU

side

<BPA>-<port> or <FSP>-<port>. The side information for the BPA/FSP. The side attribute refers to which BPA/FSP, A or B, which is determined by the slot value returned from lsslp command. It also lists the physical port within each BPA/FSP which is determined by the IP address order from the lsslp response. This information is used internally when communicating with the BPAs/FSPs

asset

A field for administators to use to correlate inventory numbers they may have to accomodate

uuid

The UUID applicable to the node

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

websrv.5

NAME

websrv - a table in the xCAT database.

SYNOPSIS

websrv Attributes: *node*, *port*, *username*, *password*, *comments*, *disable*

DESCRIPTION

Web service parameters

websrv Attributes:

node

The web service hostname.

port

The port of the web service.

username

Userid to use to access the web service.

password

Password to use to access the web service.

comments

Any user-written notes.

disable

Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

winimage.5

NAME

winimage - a table in the xCAT database.

SYNOPSIS

winimage Attributes: *imagename*, *template*, *installto*, *partitionfile*, *winpepath*, *comments*, *disable*

DESCRIPTION

Information about a Windows operating system image that can be used to deploy cluster nodes.

winimage Attributes:

imagename

The name of this xCAT OS image definition.

template

The fully qualified name of the template file that is used to create the windows unattend.xml file for diskful installation.

installto

The disk and partition that the Windows will be deployed to. The valid format is <disk>:<partition>. If not set, default value is 0:1 for bios boot mode(legacy) and 0:3 for uefi boot mode; If setting to 1, it means 1:1 for bios boot and 1:3 for uefi boot

partitionfile

The path of partition configuration file. Since the partition configuration for bios boot mode and uefi boot mode are different, this configuration file can include both configurations if you need to support both bios and uefi mode. Either way, you must specify the boot mode in the configuration. Example of partition configuration file: [BIOS]xxxxxxx[UEFI]yyyyyyy. To simplify the setting, you also can set installto in partitionfile with section like [INSTALLTO]0:1

winpepath

The path of winpe which will be used to boot this image. If the real path is /tftpboot/winboot/winpe1/, the value for winpepath should be set to winboot/winpe1

comments

> Any user-written notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

xcatdb.5

NAME

An overview of the xCAT database.

DESCRIPTION

The xCAT database contains user settings for the cluster and information gathered from the cluster. It consists of a series of tables, which are described below. To get more information about a particular table, run man for that table name. The tables can be manipulated directly using the **tabedit** or **chtab** commands. They can be viewed using **nodels** or **tabdump**.

Alternatively, the xCAT database can be viewed and edited as logical objects, instead of flat tables. In this mode, xCAT takes care of which table each attribute should go in. To treat the database as logical object definitions, use the commands: **lsdef**, **mkdef**, **chdef**, **rmdef**. See Object Definitions below.

xCAT allows the use of different database applications, depending on the needs of your cluster. The default database is SQLite, which is a daemonless, zero-config database. But you could instead choose to use something like postgresql for greater scalability and remote access in the hierarchical/service node case. To use a different database or a different location, create the file /etc/xcat/cfgloc. See the appropriate xCAT docuementation for the format of the file for the database you choose. The following example /etc/xcat/cfgloc file is for PostgreSQL:

```
Pg:dbname=xcat;host=<mgmtnode>|<pgadminuserid>|<pgadminpasswd>
```

where mgmtnode is the hostname of the management node adapter on the cluster side, and the pgadminuserid and pgadminpasswd are the database admin and password.

GROUPS AND REGULAR EXPRESSIONS IN TABLES

The xCAT database has a number of tables, some with rows that are keyed by node name (such as noderes and nodehm) and others that are not keyed by node name (for example, the policy table). The tables that are keyed by node name have some extra features that enable a more template-based style to be used:

Any group name can be used in lieu of a node name in the node field, and that row will then provide "default" attribute values for any node in that group. A row with a specific node name can then override one or more attribute values for that specific node. For example, if the nodehm table contains:

```
#node,power,mgt,cons,termserver,termport,conserver,serialport,serialspeed,serialflow,
→getmac,cmdmapping,comments,disable
"mygroup",,"ipmi",,,,,,"19200",,,,
"node1",,,,,,,,"115200",,,,
```

In the above example, the node group called mygroup sets mgt=ipmi and serialspeed=19200. Any nodes that are in this group will have those attribute values, unless overridden. For example, if node2 is a member of mygroup, it will automatically inherit these attribute values (even though it is not explicitly listed in this table). In the case of node1 above, it inherits mgt=ipmi, but overrides the serialspeed to be 115200, instead of 19200. A useful, typical way to use this capability is to create a node group for your nodes and for all the attribute values that are the same for every node, set them at the group level. Then you only have to set attributes for each node that vary from node to node.

xCAT extends the group capability so that it can also be used for attribute values that vary from node to node in a very regular pattern. For example, if in the ipmi table you want the bmc attribute to be set to whatever the nodename is with "-bmc" appended to the end of it, then use this in the ipmi table:

```
#node,bmc,bmcport,taggedvlan,bmcid,username,password,comments,disable
"compute","/\z/-bmc/",,,,,,,
```

In this example, "compute" is a node group that contains all of the compute nodes. The 2nd attribute (bmc) is a regular expression that is similar to a substitution pattern. The 1st part "z" matches the end of the node name and substitutes "-bmc", effectively appending it to the node name.

Another example is if node1 is to have IP address 10.0.0.1, node2 is to have IP address 10.0.0.2, etc., then this could be represented in the hosts table with the single row:

```
#node,ip,hostnames,otherinterfaces,comments,disable
"compute","|node(\d+)|10.0.0.($1+0)|",,,,
```

In this example, the regular expression in the ip attribute uses "|" to separate the 1st and 2nd part. This means that xCAT will allow arithmetic operations in the 2nd part. In the 1st part, "(d+)", will match the number part of the node name and put that in a variable called $1. The 2nd part is what value to give the ip attribute. In this case it will set it to the string "10.0.0." and the number that is in $1. (Zero is added to $1 just to remove any leading zeroes.)

A more involved example is with the mp table. If your blades have node names node01, node02, etc., and your chassis node names are cmm01, cmm02, etc., then you might have an mp table like:

```
#node,mpa,id,nodetype,comments,disable
"blade","|\D+(\d+)|cmm(sprintf('%02d',($1-1)/14+1))|","|\D+(\d+)|(($1-1)%14+1)|",,
```

Before you panic, let me explain each column:

blade

> This is a group name. In this example, we are assuming that all of your blades belong to this group. Each time the xCAT software accesses the **mp** table to get the management module and slot number of a specific blade (e.g. **node20**), this row will match (because **node20** is in the **blade** group). Once this row is matched for **node20**, then the processing described in the following items will take place.

|D+(d+)|cmm(sprintf('%02d',($1-1)/14+1))|

> This is a perl substitution pattern that will produce the value for the second column of the table (the management module hostname). The text **D+(d+)** between the 1st two vertical bars is a regular expression that matches the node name that was searched for in this table (in this example **node20**). The text that matches within the 1st set of parentheses is set to $1. (If there was a 2nd set of parentheses, it would be set to $2, and so on.) In our case, the D+ matches the non-numeric part of the name (**node**) and the **d+** matches the numeric part (**20**). So $1 is set to **20**. The text **cmm(sprintf('%02d',($1-1)/14+1))** between the 2nd and 3rd vertical bars produces the string that should be used as the value for the mpa attribute

for node20. Since $1 is set to 20, the expression **($1-1)/14+1** equals 19/14 + 1, which equals 2. (The division is integer division, so 19/14 equals 1. Fourteen is used as the divisor, because there are 14 blades in each chassis.) The value of 2 is then passed into sprintf() with a format string to add a leading zero, if necessary, to always make the number two digits. Lastly the string **cmm** is added to the beginning, making the resulting string **cmm02**, which will be used as the hostname of the management module.

|D+(d+)|(($1-1)%14+1)|

This item is similar to the one above. This substituion pattern will produce the value for the 3rd column (the chassis slot number for this blade). Because this row was the match for **node20**, the parentheses within the 1st set of vertical bars will set $1 to 20. Since % means modulo division, the expression **($1-1)%14+1** will evaluate to **6**.

See http://www.perl.com/doc/manual/html/pod/perlre.html for information on perl regular expressions.

Easy Regular Expressions

As of xCAT 2.8.1, you can use a modified version of the regular expression support described in the previous section. You do not need to enter the node information (1st part of the expression), it will be derived from the input nodename. You only need to supply the 2nd part of the expression to determine the value to give the attribute. For examples, see

https://sourceforge.net/p/xcat/wiki/Listing_and_Modifying_the_Database/#easy-regular-expressions

OBJECT DEFINITIONS

Because it can get confusing what attributes need to go in what tables, the xCAT database can also be viewed and edited as logical objects, instead of flat tables. Use **mkdef**, **chdef**, **lsdef**, and **rmdef** to create, change, list, and delete objects. When using these commands, the object attributes will be stored in the same tables, as if you edited the tables by hand. The only difference is that the object commands take care of knowing which tables all of the information should go in.

To run man for any of the object definitions below, use section 7. For example: **man 7 node**

The object types are:

auditlog(7)|auditlog.7

boottarget(7)|boottarget.7

eventlog(7)|eventlog.7

firmware(7)|firmware.7

group(7)|group.7

kit(7)|kit.7

kitcomponent(7)|kitcomponent.7

kitrepo(7)|kitrepo.7

monitoring(7)|monitoring.7

network(7)|network.7

node(7)|node.7

notification(7)|notification.7

osdistro(7)|osdistro.7

osdistroupdate(7)|osdistroupdate.7

osimage(7)|osimage.7

policy(7)|policy.7

rack(7)|rack.7

route(7)|route.7

site(7)|site.7

zone(7)|zone.7

TABLES

To manipulate the tables directly, use **nodels(1)**, **chtab(8)**, **tabdump(8)**, **tabedit(8)**, **nodeadd(8)**, **nodech(1)**.

To run man for any of the table descriptions below, use section 5. For example: **man 5 nodehm**

The tables are:

auditlog(5)|auditlog.5

```
Audit Data log.
```

bootparams(5)|bootparams.5

> Current boot settings to be sent to systems attempting network boot for deployment, stateless, or other reasons. Mostly automatically manipulated by xCAT.

boottarget(5)|boottarget.5

> Specify non-standard initrd, kernel, and parameters that should be used for a given profile.

cfgmgt(5)|cfgmgt.5

> Configuration management data for nodes used by non-xCAT osimage management services to install and configure software on a node.

chain(5)|chain.5

> Controls what operations are done (and it what order) when a node is discovered and deployed.

deps(5)|deps.5

> Describes dependencies some nodes have on others. This can be used, e.g., by rpower -d to power nodes on or off in the correct order.

discoverydata(5)|discoverydata.5

> Discovery data which sent from genesis.

domain(5)|domain.5

> Mapping of nodes to domain attributes

eventlog(5)|eventlog.5

> Stores the events occurred.

firmware(5)|firmware.5

> Maps node to firmware values to be used for setup at node discovery or later

hosts(5)|hosts.5

IP addresses and hostnames of nodes. This info is optional and is only used to populate /etc/hosts and DNS via makehosts and makedns. Using regular expressions in this table can be a quick way to populate /etc/hosts.

hwinv(5)|hwinv.5

The hareware inventory for the node.

hypervisor(5)|hypervisor.5

Hypervisor parameters

ipmi(5)|ipmi.5

Settings for nodes that are controlled by an on-board BMC via IPMI.

iscsi(5)|iscsi.5

Contains settings that control how to boot a node from an iSCSI target

kit(5)|kit.5

This table stores all kits added to the xCAT cluster.

kitcomponent(5)|kitcomponent.5

This table stores all kit components added to the xCAT cluster.

kitrepo(5)|kitrepo.5

This table stores all kits added to the xCAT cluster.

kvm_masterdata(5)|kvm_masterdata.5

Persistant store for KVM plugin for masters

kvm_nodedata(5)|kvm_nodedata.5

Persistant store for KVM plugin, not intended for manual modification.

linuximage(5)|linuximage.5

Information about a Linux operating system image that can be used to deploy cluster nodes.

litefile(5)|litefile.5

The litefile table specifies the directories and files on the statelite nodes that should be readwrite, persistent, or readonly overlay. All other files in the statelite nodes come from the readonly statelite image.

litetree(5)|litetree.5

Directory hierarchy to traverse to get the initial contents of node files. The files that are specified in the litefile table are searched for in the directories specified in this table.

mac(5)|mac.5

The MAC address of the node's install adapter. Normally this table is populated by getmacs or node discovery, but you can also add entries to it manually.

mic(5)|mic.5

The host, slot id and configuraton of the mic (Many Integrated Core).

monitoring(5)|monitoring.5

Controls what external monitoring tools xCAT sets up and uses. Entries should be added and removed from this table using the provided xCAT commands monstart and monstop.

monsetting(5)|monsetting.5

Specifies the monitoring plug-in specific settings. These settings will be used by the monitoring plug-in to customize the behavior such as event filter, sample interval, responses etc. Entries should be added, removed or modified by chtab command. Entries can also be added or modified by the monstart command when a monitoring plug-in is brought up.

mp(5)|mp.5

Contains the hardware control info specific to blades. This table also refers to the mpa table, which contains info about each Management Module.

mpa(5)|mpa.5

Contains info about each Management Module and how to access it.

networks(5)|networks.5

Describes the networks in the cluster and info necessary to set up nodes on that network.

nics(5)|nics.5

Stores NIC details.

nimimage(5)|nimimage.5

All the info that specifies a particular AIX operating system image that can be used to deploy AIX nodes.

nodegroup(5)|nodegroup.5

Contains group definitions, whose membership is dynamic depending on characteristics of the node.

nodehm(5)|nodehm.5

Settings that control how each node's hardware is managed. Typically, an additional table that is specific to the hardware type of the node contains additional info. E.g. the ipmi, mp, and ppc tables.

nodelist(5)|nodelist.5

The list of all the nodes in the cluster, including each node's current status and what groups it is in.

nodepos(5)|nodepos.5

Contains info about the physical location of each node. Currently, this info is not used by xCAT, and therefore can be in whatevery format you want. It will likely be used in xCAT in the future.

noderes(5)|noderes.5

Resources and settings to use when installing nodes.

nodetype(5)|nodetype.5

A few hardware and software characteristics of the nodes.

notification(5)|notification.5

Contains registrations to be notified when a table in the xCAT database changes. Users can add entries to have additional software notified of changes. Add and remove entries using the provided xCAT commands regnotif and unregnotif.

osdistro(5)|osdistro.5

Information about all the OS distros in the xCAT cluster

osdistroupdate(5)|osdistroupdate.5

Information about the OS distro updates in the xCAT cluster

osimage(5)|osimage.5

Basic information about an operating system image that can be used to deploy cluster nodes.

passwd(5)|passwd.5

> Contains default userids and passwords for xCAT to access cluster components. In most cases, xCAT will also actually set the userid/password in the relevant component when it is being configured or installed. Userids/passwords for specific cluster components can be overidden in other tables, e.g. mpa, ipmi, ppchcp, etc.

performance(5)|performance.5

> Describes the system performance every interval unit of time.

policy(5)|policy.5

> The policy table in the xCAT database controls who has authority to run specific xCAT operations. It is basically the Access Control List (ACL) for xCAT. It is sorted on the priority field before evaluating.

postscripts(5)|postscripts.5

> The scripts that should be run on each node after installation or diskless boot.

ppc(5)|ppc.5

> List of system p hardware: HMCs, IVMs, FSPs, BPCs, CECs, Frames.

ppcdirect(5)|ppcdirect.5

> Info necessary to use FSPs/BPAs to control system p CECs/Frames.

ppchcp(5)|ppchcp.5

> Info necessary to use HMCs and IVMs as hardware control points for LPARs.

prescripts(5)|prescripts.5

> The scripts that will be run at the beginning and the end of the nodeset(Linux), nimnodeset(AIX) or mkdsklsnode(AIX) command.

prodkey(5)|prodkey.5

> Specify product keys for products that require them

rack(5)|rack.5

> Rack information.

routes(5)|routes.5

> Describes the additional routes needed to be setup in the os routing table. These routes usually are used to connect the management node to the compute node using the servie node as gateway.

servicenode(5)|servicenode.5

> List of all Service Nodes and services that will be set up on the Service Node.

site(5)|site.5

> Global settings for the whole cluster. This table is different from the other tables in that each attribute is just named in the key column, rather than having a separate column for each attribute. The following is a list of attributes currently used by xCAT organized into categories.

statelite(5)|statelite.5

> The location on an NFS server where a nodes persistent files are stored. Any file marked persistent in the litefile table will be stored in the location specified in this table for that node.

storage(5)|storage.5

switch(5)|switch.5

Contains what switch port numbers each node is connected to.

switches(5)|switches.5

Parameters to use when interrogating switches

token(5)|token.5

The token of users for authentication.

virtsd(5)|virtsd.5

The parameters which used to create the Storage Domain

vm(5)|vm.5

Virtualization parameters

vmmaster(5)|vmmaster.5

Inventory of virtualization images for use with clonevm. Manual intervention in this table is not intended.

vpd(5)|vpd.5

The Machine type, Model, and Serial numbers of each node.

websrv(5)|websrv.5

Web service parameters

winimage(5)|winimage.5

Information about a Windows operating system image that can be used to deploy cluster nodes.

zone(5)|zone.5

Defines a cluster zone for nodes that share root ssh key access to each other.

zvm(5)|zvm.5

List of z/VM virtual servers.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**, **lsdef(1)**, **mkdef(1)**, **chdef(1)**, **rmdef(1)**

xcatstanzafile.5

NAME

xcatstanzafile - Format of a stanza file that can be used with xCAT data object definition commands.

DESCRIPTION

A stanza file contains information that can be used to create xCAT data object definitions. A stanza file can be used as input to several xCAT commands. The stanza file contains one or more individual stanzas that provide information for individual object definitions. The following rules must be followed when creating a stanza file:

*

An object stanza header consists of the object name followed by a colon, (":").

*

   Attribute lines must take the form of Attribute=Value.

*

   Attribute name might include the character dot ("."), like passwd.HMC and nicips.eth0.

*

   Only one stanza can exist for each object name.

*

   All stanzas except for default stanzas must have a value set for "objtype".

*

   Comments beginning with the "#" pound sign may be added to the file. A comment must be on a separate line.

*

   When parsing the file, tab characters and spaces are ignored.

*

   Each line of the file can have no more than one header or attribute definition.

*

   If the header name is "default-<object type>:" the attribute values in the stanza are considered default values for subsequent definitions in the file that are the same object type.

*

   Default stanzas can be specified multiple times and at any point in a stanza file. The values apply to all definitions following the default stanzas in a file. The default values are cumulative; a default attribute value will remain set until it is explicitly unset or changed.

*

   To turn off a default value, use another default stanza to set the attribute to have no value using a blank space.

*

   When a specific value for an attribute is provided in the stanza, it takes priority over any default value that had been set.

The format of a stanza file should look similar to the following.

```
default-<object type>:
   attr=val
   attr=val
   . . .

<object name>:
   objtype=<object type>
   attr=val
   attr=val
   . . .

<object name>:
   objtype=<object type>
   attr=val
```

```
   attr=val
   . . .
```

EXAMPLES

1. Sample stanza file:

```
mysite:
    objtype=site
    rsh=/bin/rsh
    rcp=/bin/rcp
    installdir=/xcatinstall
    domain=ppd.pok.ibm.com

MSnet01:
    objtype=network
    gateway=1.2.3.4
    netmask=255.255.255.0
    nameserver=5.6.7.8

default-node:
    next_osimage=aix61
    network=MSnet01
    groups=all,compute

node01:
    objtype=node
    MAC=A2E26002C003
    xcatmaster=MS02.ppd.pok.com
    nfsserver=IS227.ppd.pok.com

node02:
    objtype=node
    MAC=A2E26002B004
    xcatmaster=MS01.ppd.pok.com
    nfsserver=IS127.ppd.pok.com

grp01:
    objtype=group
    members=node1,node2,node3
```

NOTES

This file is part of xCAT software product.

SEE ALSO

mkdef(1)|mkdef.1, lsdef(1)|lsdef.1, rmdef(1)|rmdef.1, chdef(1)|chdef.1

zone.5

NAME

zone - a table in the xCAT database.

SYNOPSIS

zone Attributes: *zonename*, *sshkeydir*, *sshbetweennodes*, *defaultzone*, *comments*, *disable*

DESCRIPTION

Defines a cluster zone for nodes that share root ssh key access to each other.

zone Attributes:

zonename

> The name of the zone.

sshkeydir

> Directory containing the shared root ssh RSA keys.

sshbetweennodes

> Indicates whether passwordless ssh will be setup between the nodes of this zone. Values are yes/1 or no/0. Default is yes.

defaultzone

> If nodes are not assigned to any other zone, they will default to this zone. If value is set to yes or 1.

comments

> Any user-provided notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

zvm.5

NAME

zvm - a table in the xCAT database.

SYNOPSIS

zvm Attributes: *node*, *hcp*, *userid*, *nodetype*, *parent*, *comments*, *disable*

DESCRIPTION

List of z/VM virtual servers.

zvm Attributes:

node

> The node name.

hcp

> The hardware control point for this node.

userid

> The z/VM userID of this node.

nodetype

> The node type. Valid values: cec (Central Electronic Complex), lpar (logical partition), zvm (z/VM host operating system), and vm (virtual machine).

parent

> The parent node. For LPAR, this specifies the CEC. For z/VM, this specifies the LPAR. For VM, this specifies the z/VM host operating system.

comments

> Any user provided notes.

disable

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

nodels(1), **chtab(8)**, **tabdump(8)**, **tabedit(8)**

man7

auditlog.7

NAME

auditlog - a logical object definition in the xCAT database.

SYNOPSIS

auditlog Attributes: *args*, *audittime*, *clientname*, *clienttype*, *command*, *comments*, *disable*, *noderange*, *recid*, *status*, *userid*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

auditlog Attributes:

args (auditlog.args)

> The command argument list.

audittime (auditlog.audittime)

> The timestamp for the audit entry.

clientname (auditlog.clientname)

> The client machine, where the command originated.

clienttype (auditlog.clienttype)

> Type of command: cli,java,webui,other.

command (auditlog.command)

> Command executed.

comments (auditlog.comments)

> Any user-provided notes.

disable (auditlog.disable)

> Do not use. tabprune will not work if set to yes or 1

noderange (auditlog.noderange)

> The noderange on which the command was run.

recid (auditlog.recid)

> The record id.

status (auditlog.status)

> Allowed or Denied.

userid (auditlog.userid)

> The user running the command.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

boottarget.7

NAME

boottarget - a logical object definition in the xCAT database.

SYNOPSIS

boottarget Attributes: **

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

boottarget Attributes:

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

eventlog.7

NAME

eventlog - a logical object definition in the xCAT database.

SYNOPSIS

eventlog Attributes: *application*, *comments*, *component*, *disable*, *eventtime*, *eventtype*, *id*, *message*, *monitor*, *monnode*, *node*, *rawdata*, *recid*, *severity*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

eventlog Attributes:

application (eventlog.application)

> The application that reports the event.

comments (eventlog.comments)

>   Any user-provided notes.

component (eventlog.component)

>   The component where the event occurred.

disable (eventlog.disable)

>   Do not use. tabprune will not work if set to yes or 1

eventtime (eventlog.eventtime)

>   The timestamp for the event.

eventtype (eventlog.eventtype)

>   The type of the event.

id (eventlog.id)

>   The location or the resource name where the event occurred.

message (eventlog.message)

>   The full description of the event.

monitor (eventlog.monitor)

>   The name of the monitor that monitors this event.

monnode (eventlog.monnode)

>   The node that monitors this event.

node (eventlog.node)

>   The node where the event occurred.

rawdata (eventlog.rawdata)

```
The data that associated with the event.
```

recid (eventlog.recid)

>   The record id.

severity (eventlog.severity)

>   The severity of the event. Valid values are: informational, warning, critical.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

firmware.7

NAME

firmware - a logical object definition in the xCAT database.

SYNOPSIS

firmware Attributes: *cfgfile*, *comments*, *disable*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

firmware Attributes:

cfgfile (firmware.cfgfile)

> The file to use.

comments (firmware.comments)

> Any user-written notes.

disable (firmware.disable)

> Set to 'yes' or '1' to comment out this row.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

group.7

NAME

group - a logical object definition in the xCAT database.

SYNOPSIS

group Attributes: *addkcmdline, arch, authdomain, bmc, bmcpassword, bmcport, bmcusername, cfgmgr, cfgmgtroles, cfgserver, chain, chassis, cmdmapping, cons, conserver, consoleondemand, cpucount, cputype, currchain, currstate, dhcpinterfaces, disksize, displayname, domainadminpassword, domainadminuser, domaintype, getmac, groupname, grouptype, hcp, height, hostcluster, hostinterface, hostmanager, hostnames, hosttype, hwtype, id, initrd, installnic, interface, ip, iscsipassword, iscsiserver, iscsitarget, iscsiuserid, kcmdline, kernel, mac, membergroups, members, memory, mgt, micbridge, michost, micid, miconboot, micpowermgt, micvlog, migrationdest, monserver, mpa, mtm, nameservers, netboot, nfsdir, nfsserver, nicaliases, niccustomscripts, nicextraparams, nichostnameprefixes, nichostnamesuffixes, nicips, nicnetworks, nictypes, nimserver, nodetype, ondiscover, os, osvolume, otherinterfaces, ou, parent, passwd.HMC, passwd.admin, passwd.celogin, passwd.general, passwd.hscroot, password, postbootscripts, postscripts, power, pprofile, prescripts-begin, prescripts-end, primarynic, productkey, profile, provmethod, rack, room, routenames, serial, serialflow, serialport, serialspeed, servicenode, setupconserver, setupdhcp, setupftp, setupipforward, setupldap, setupnameserver, setupnfs, setupnim, setupntp, setupproxydhcp, setuptftp, sfp, side, slot, slotid, slots, storagcontroller, storagetype, supernode, supportedarchs, supportproxydhcp, switch, switchinterface, switchport, switchvlan, termport, termserver, tftpdir, tftpserver, unit, urlpath, usercomment, userid, username, vmbeacon,*

vmbootorder, *vmcfgstore*, *vmcluster*, *vmcpus*, *vmhost*, *vmmanager*, *vmmaster*, *vmmemory*, *vmnicnicmodel*, *vmnics*, *vmothersetting*, *vmphyslots*, *vmstorage*, *vmstoragecache*, *vmstorageformat*, *vmstoragemodel*, *vmtextconsole*, *vmvirt-flags*, *vmvncport*, *webport*, *wherevals*, *xcatmaster*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

group Attributes:

addkcmdline (bootparams.addkcmdline)

> User specified one or more parameters to be passed to the kernel. For the kernel options need to be persistent after installation, specify them with prefix "R::"

arch (nodetype.arch)

> The hardware architecture of this node. Valid values: x86_64, ppc64, x86, ia64.

authdomain (domain.authdomain)

> If a node should participate in an AD domain or Kerberos realm distinct from domain indicated in site, this field can be used to specify that

bmc (ipmi.bmc)

> The hostname of the BMC adapater.

bmcpassword (ipmi.password)

> The BMC password. If not specified, the key=ipmi row in the passwd table is used as the default.

bmcport (ipmi.bmcport)

```
In systems with selectable shared/dedicated ethernet ports,

        this parameter can be used to specify the preferred port.  0

        means use the shared port, 1 means dedicated, blank is to not

        assign.


        The following special cases exist for IBM System x servers:


        For x3755 M3 systems, 0 means use the dedicated port, 1 means

        shared, blank is to not assign.


    For certain systems which have a mezzaine or ML2 adapter, there is a␣
→second
```

```
    value to include:



    For x3750 M4 (Model 8722):



    0 2   1st 1Gbps interface for LOM


    0 0   1st 10Gbps interface for LOM


    0 3   2nd 1Gbps interface for LOM


    0 1   2nd 10Gbps interface for LOM



    For  x3750 M4 (Model 8752), x3850/3950 X6, dx360 M4, x3550 M4, and
→x3650 M4:



    0     Shared (1st onboard interface)


    1     Dedicated


    2 0   First interface on ML2 or mezzanine adapter


    2 1   Second interface on ML2 or mezzanine adapter


    2 2   Third interface on ML2 or mezzanine adapter
```

```
        2  3    Fourth interface on ML2 or mezzanine adapter
```

bmcusername (ipmi.username)

>   The BMC userid. If not specified, the key=ipmi row in the passwd table is used as the default.

cfgmgr (cfgmgt.cfgmgr)

>   The name of the configuration manager service. Currently 'chef' and 'puppet' are supported services.

cfgmgtroles (cfgmgt.roles)

>   The roles associated with this node as recognized by the cfgmgr for the software that is to be installed and configured. These role names map to chef recipes or puppet manifest classes that should be used for this node. For example, chef OpenStack cookbooks have roles such as mysql-master,keystone, glance, nova-controller, nova-conductor, cinder-all.

cfgserver (cfgmgt.cfgserver)

>   The xCAT node name of the chef server or puppet master

chain (chain.chain)

>   A comma-delimited chain of actions to be performed automatically when this node is discovered. ("Discovered" means a node booted, but xCAT and DHCP did not recognize the MAC of this node. In this situation, xCAT initiates the discovery process, the last step of which is to run the operations listed in this chain attribute, one by one.) Valid values: boot or reboot, install or netboot, runcmd=<cmd>, runimage=<URL>, shell, standby. (Default - same as no chain - it will do only the discovery.). Example, for BMC machines use: runcmd=bmcsetup,shell.

chassis (nodepos.chassis)

>   The BladeCenter chassis the blade is in.

cmdmapping (nodehm.cmdmapping)

>   The fully qualified name of the file that stores the mapping between PCM hardware management commands and xCAT/third-party hardware management commands for a particular type of hardware device. Only used by PCM.

cons (nodehm.cons)

>   The console method. If nodehm.serialport is set, this will default to the nodehm.mgt setting, otherwise it defaults to unused. Valid values: cyclades, mrv, or the values valid for the mgt attribute.

conserver (nodehm.conserver)

>   The hostname of the machine where the conserver daemon is running. If not set, the default is the xCAT management node.

consoleondemand (nodehm.consoleondemand)

>   This overrides the value from site.consoleondemand; (0=no, 1=yes). Default is the result from site.consoleondemand.

cpucount (hwinv.cpucount)

>   The number of cpus for the node.

cputype (hwinv.cputype)

>   The cpu model name for the node.

currchain (chain.currchain)

The chain steps still left to do for this node. This attribute will be automatically adjusted by xCAT while xCAT-genesis is running on the node (either during node discovery or a special operation like firmware update). During node discovery, this attribute is initialized from the chain attribute and updated as the chain steps are executed.

currstate (chain.currstate)

The current or next chain step to be executed on this node by xCAT-genesis. Set by xCAT during node discovery or as a result of nodeset.

dhcpinterfaces (servicenode.dhcpinterfaces)

The network interfaces DHCP server should listen on for the target node. This attribute can be used for management node and service nodes. If defined, it will override the values defined in site.dhcpinterfaces. This is a comma separated list of device names. !remote! indicates a non-local network for relay DHCP. For example: !remote!,eth0,eth1

disksize (hwinv.disksize)

The size of the disks for the node in GB.

displayname (mpa.displayname)

Alternative name for BladeCenter chassis. Only used by PCM.

domainadminpassword (domain.adminpassword)

Allow a node specific indication of Administrative user password for the domain. Most will want to ignore this in favor of passwd table.

domainadminuser (domain.adminuser)

Allow a node specific indication of Administrative user. Most will want to just use passwd table to indicate this once rather than by node.

domaintype (domain.type)

Type, if any, of authentication domain to manipulate. The only recognized value at the moment is activedirectory.

getmac (nodehm.getmac)

The method to use to get MAC address of the node with the getmac command. If not set, the mgt attribute will be used. Valid values: same as values for mgmt attribute.

groupname (nodegroup.groupname)

Name of the group.

grouptype (nodegroup.grouptype)

The only current valid value is dynamic. We will be looking at having the object def commands working with static group definitions in the nodelist table.

hcp (ppc.hcp, zvm.hcp)

The hardware control point for this node (HMC, IVM, Frame or CEC). Do not need to set for BPAs and FSPs.

or

The hardware control point for this node.

height (nodepos.height)

The server height in U(s).

hostcluster (hypervisor.cluster)

Specify to the underlying virtualization infrastructure a cluster membership for the hypervisor.

hostinterface (hypervisor.interface)

The definition of interfaces for the hypervisor. The format is [network-name:interfacename:bootprotocol:IP:netmask:gateway] that split with | for each interface

hostmanager (hypervisor.mgr)

The virtualization specific manager of this hypervisor when applicable

hostnames (hosts.hostnames)

Hostname aliases added to /etc/hosts for this node. Comma or blank separated list.

hosttype (hypervisor.type)

The plugin associated with hypervisor specific commands such as revacuate

hwtype (ppc.nodetype, zvm.nodetype, mp.nodetype, mic.nodetype)

The hardware type of the node. Only can be one of fsp, bpa, cec, frame, ivm, hmc and lpar

or

The node type. Valid values: cec (Central Electronic Complex), lpar (logical partition), zvm (z/VM host operating system), and vm (virtual machine).

or

The hardware type for mp node. Valid values: mm,cmm, blade.

or

The hardware type of the mic node. Generally, it is mic.

id (ppc.id, mp.id)

For LPARs: the LPAR numeric id; for CECs: the cage number; for Frames: the frame number.

or

The slot number of this blade in the BladeCenter chassis.

initrd (bootparams.initrd)

The initial ramdisk image that network boot actions should use (could be a DOS floppy or hard drive image if using memdisk as kernel)

installnic (noderes.installnic)

The network adapter on the node that will be used for OS deployment, the installnic can be set to the network adapter name or the mac address or the keyword "mac" which means that the network interface specified by the mac address in the mac table will be used. If not set, primarynic will be used. If primarynic is not set too, the keyword "mac" will be used as default.

interface (mac.interface)

The adapter interface name that will be used to install and manage the node. E.g. eth0 (for linux) or en0 (for AIX).)

ip (hosts.ip)

The IP address of the node. This is only used in makehosts. The rest of xCAT uses system name resolution to resolve node names to IP addresses.

iscsipassword (iscsi.passwd)

The password for the iscsi server containing the boot device for this node.

iscsiserver (iscsi.server)

The server containing the iscsi boot device for this node.

iscsitarget (iscsi.target)

The iscsi disk used for the boot device for this node. Filled in by xCAT.

iscsiuserid (iscsi.userid)

The userid of the iscsi server containing the boot device for this node.

kcmdline (bootparams.kcmdline)

Arguments to be passed to the kernel

kernel (bootparams.kernel)

The kernel that network boot actions should currently acquire and use. Note this could be a chained boot loader such as memdisk or a non-linux boot loader

mac (mac.mac)

The mac address or addresses for which xCAT will manage static bindings for this node. This may be simply a mac address, which would be bound to the node name (such as "01:02:03:04:05:0E"). This may also be a "|" delimited string of "mac address!hostname" format (such as "01:02:03:04:05:0E!node5|01:02:03:05:0F!node6-eth1").

membergroups (nodegroup.membergroups)

This attribute stores a comma-separated list of nodegroups that this nodegroup refers to. This attribute is only used by PCM.

members (nodegroup.members)

The value of the attribute is not used, but the attribute is necessary as a place holder for the object def commands. (The membership for static groups is stored in the nodelist table.)

memory (hwinv.memory)

The size of the memory for the node in MB.

mgt (nodehm.mgt)

The method to use to do general hardware management of the node. This attribute is used as the default if power or getmac is not set. Valid values: ipmi, blade, hmc, ivm, fsp, bpa, kvm, esx, rhevm. See the power attribute for more details.

micbridge (mic.bridge)

The virtual bridge on the host node which the mic connected to.

michost (mic.host)

The host node which the mic card installed on.

micid (mic.id)

The device id of the mic node.

miconboot (mic.onboot)

Set mic to autoboot when mpss start. Valid values: yes|no. Default is yes.

micpowermgt (mic.powermgt)

Set the Power Management for mic node. This attribute is used to set the power management state that mic may get into when it is idle. Four states can be set: cpufreq, corec6, pc3 and pc6. The valid value for powermgt attribute should be [cpufreq=<on|off>]![corec6=<on|off>]![pc3=<on|off>]![pc6=<on|off>]. e.g. cpufreq=on!corec6=off!pc3=on!pc6=off. Refer to the doc of mic to get more information for power management.

micvlog (mic.vlog)

Set the Verbose Log to console. Valid values: yes|no. Default is no.

migrationdest (vm.migrationdest)

A noderange representing candidate destinations for migration (i.e. similar systems, same SAN, or other criteria that xCAT can use

monserver (noderes.monserver)

The monitoring aggregation point for this node. The format is "x,y" where x is the ip address as known by the management node and y is the ip address as known by the node.

mpa (mp.mpa)

The managment module used to control this blade.

mtm (vpd.mtm)

The machine type and model number of the node. E.g. 7984-6BU

nameservers (noderes.nameservers)

An optional node/group specific override for name server list. Most people want to stick to site or network defined nameserver configuration.

netboot (noderes.netboot)

The type of network booting to use for this node. Valid values: pxe or xnba for x86* architecture, yaboot for POWER architecture, grub2-tftp and grub2-http for RHEL7 on Power and all the os deployment on Power LE. Notice: yaboot is not supported from rhels7 on Power,use grub2-tftp or grub2-http instead, the difference between the 2 is the file transfer protocol(i.e, http or tftp)

nfsdir (noderes.nfsdir)

The path that should be mounted from the NFS server.

nfsserver (noderes.nfsserver)

The NFS or HTTP server for this node (as known by this node).

nicaliases (nics.nicaliases)

Comma-separated list of hostname aliases for each NIC.

```
Format: eth0!<alias list>,eth1!<alias1 list>|<alias2 list>

        For multiple aliases per nic use a space-separated list.

For example: eth0!moe larry curly,eth1!tom|jerry
```

niccustomscripts (nics.niccustomscripts)

Comma-separated list of custom scripts per NIC. <nic1>!<script1>,<nic2>!<script2>, e.g. eth0!configeth eth0, ib0!configib ib0. The xCAT object definition commands support to use niccustomscripts.<nicname> as the sub attribute

.

nicextraparams (nics.nicextraparams)

Comma-separated list of extra parameters that will be used for each NIC configuration.

```
    If only one ip address is associated with each NIC:

        <nic1>!<param1=value1 param2=value2>,<nic2>!<param3=value3>, for
→example, eth0!MTU=1500,ib0!MTU=65520 CONNECTED_MODE=yes.

    If multiple ip addresses are associated with each NIC:

        <nic1>!<param1=value1 param2=value2>|<param3=value3>,<nic2>!
→<param4=value4 param5=value5>|<param6=value6>, for example, eth0!
→MTU=1500|MTU=1460,ib0!MTU=65520 CONNECTED_MODE=yes.

The xCAT object definition commands support to use nicextraparams.<nicname>
→as the sub attributes.
```

nichostnameprefixes (nics.nichostnameprefixes)

Comma-separated list of hostname prefixes per NIC.

```
If only one ip address is associated with each NIC:

    <nic1>!<ext1>,<nic2>!<ext2>,..., for example, eth0!eth0-,ib0!ib-

If multiple ip addresses are associcated with each NIC:

    <nic1>!<ext1>|<ext2>,<nic2>!<ext1>|<ext2>,..., for example,  eth0!eth0-
→|eth0-ipv6i-,ib0!ib-|ib-ipv6-.

The xCAT object definition commands support to use nichostnameprefixes.
→<nicname> as the sub attributes.

Note:  According to DNS rules a hostname must be a text string up to 24
→characters drawn from the alphabet (A-Z), digits (0-9), minus sign (-),and
→period (.). When you are specifying "nichostnameprefixes" or "nicaliases"
→make sure the resulting hostnames will conform to this naming convention
```

nichostnamesuffixes (nics.nichostnamesuffixes)

Comma-separated list of hostname suffixes per NIC.

```
If only one ip address is associated with each NIC:

    <nic1>!<ext1>,<nic2>!<ext2>,..., for example, eth0!-eth0,ib0!-ib0

If multiple ip addresses are associcated with each NIC:

    <nic1>!<ext1>|<ext2>,<nic2>!<ext1>|<ext2>,..., for example,  eth0!-eth0|-
→eth0-ipv6,ib0!-ib0|-ib0-ipv6.

The xCAT object definition commands support to use nichostnamesuffixes.
→<nicname> as the sub attributes.

Note:  According to DNS rules a hostname must be a text string up to 24
→characters drawn from the alphabet (A-Z), digits (0-9), minus sign (-),and
→period (.). When you are specifying "nichostnamesuffixes" or "nicaliases"
→make sure the resulting hostnames will conform to this naming convention
```

nicips (nics.nicips)

Comma-separated list of IP addresses per NIC. To specify one ip address per NIC:

```
    <nic1>!<ip1>,<nic2>!<ip2>,..., for example, eth0!10.0.0.100,ib0!11.0.0.
↪100

To specify multiple ip addresses per NIC:

    <nic1>!<ip1>|<ip2>,<nic2>!<ip1>|<ip2>,..., for example, eth0!10.0.0.
↪100|fd55::214:5eff:fe15:849b,ib0!11.0.0.100|2001::214:5eff:fe15:849a. The␣
↪xCAT object definition commands support to use nicips.<nicname> as the sub␣
↪attributes.

Note: The primary IP address must also be stored in the hosts.ip attribute.␣
↪The nichostnamesuffixes should specify one hostname suffix for each ip␣
↪address.
```

nicnetworks (nics.nicnetworks)

Comma-separated list of networks connected to each NIC.

```
If only one ip address is associated with each NIC:

    <nic1>!<network1>,<nic2>!<network2>, for example, eth0!10_0_0_0-255_255_
↪0_0, ib0!11_0_0_0-255_255_0_0

If multiple ip addresses are associated with each NIC:

    <nic1>!<network1>|<network2>,<nic2>!<network1>|<network2>, for example,␣
↪eth0!10_0_0_0-255_255_0_0|fd55:faaf:e1ab:336::/64,ib0!11_0_0_0-255_255_0_
↪0|2001:db8:1:0::/64. The xCAT object definition commands support to use␣
↪nicnetworks.<nicname> as the sub attributes.
```

nictypes (nics.nictypes)

Comma-separated list of NIC types per NIC. <nic1>!<type1>,<nic2>!<type2>, e.g. eth0!Ethernet,ib0!Infiniband. The xCAT object definition commands support to use nictypes.<nicname> as the sub attributes.

nimserver (noderes.nimserver)

Not used for now. The NIM server for this node (as known by this node).

nodetype (nodetype.nodetype)

A comma-delimited list of characteristics of this node. Valid values: ppc, blade, vm (virtual machine), osi (OS image), mm, mn, rsa, switch.

ondiscover (chain.ondiscover)

This attribute is currently not used by xCAT. The "nodediscover" operation is always done during node discovery.

os (nodetype.os)

The operating system deployed on this node. Valid values: AIX, rhels*,rhelc*, rhas*,centos*,SL*, fedora*, sles* (where * is the version #). As a special case, if this is set to "boottarget", then it will use the initrd/kernel/parameters specified in the row in the boottarget table in which boottarget.bprofile equals nodetype.profile.

osvolume (storage.osvolume)

Specification of what storage to place the node OS image onto. Examples include:

```
localdisk (Install to first non-FC attached disk)

usbdisk (Install to first USB mass storage device seen)

wwn=0x50000393c813840c (Install to storage device with given WWN)
```

otherinterfaces (hosts.otherinterfaces)

Other IP addresses to add for this node. Format: -<ext>:<ip>,<intfhostname>:<ip>,...

ou (domain.ou)

For an LDAP described machine account (i.e. Active Directory), the orginaztional unit to place the system. If not set, defaults to cn=Computers,dc=your,dc=domain

parent (ppc.parent)

For LPARs: the CEC; for FSPs: the CEC; for CEC: the frame (if one exists); for BPA: the frame; for frame: the building block number (which consists 1 or more service nodes and compute/storage nodes that are serviced by them - optional).

passwd.HMC (ppcdirect.password)

Password of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

passwd.admin (ppcdirect.password)

Password of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

passwd.celogin (ppcdirect.password)

Password of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

passwd.general (ppcdirect.password)

Password of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

passwd.hscroot (ppcdirect.password)

Password of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

password (ppchcp.password, mpa.password, websrv.password)

Password of the HMC or IVM. If not filled in, xCAT will look in the passwd table for key=hmc or key=ivm. If not in the passwd table, the default used is abc123 for HMCs and padmin for IVMs.

or

Password to use to access the management module. If not specified, the key=blade row in the passwd table is used as the default.

or

Password to use to access the web service.

postbootscripts (postscripts.postbootscripts)

Comma separated list of scripts that should be run on this node after diskfull installation or diskless boot. Each script can take zero or more parameters. For example: "script1 p1 p2,script2,...". On AIX these scripts are run during the processing of /etc/inittab. On Linux they are run at the init.d time. xCAT automatically adds the scripts in the xcatdefaults.postbootscripts attribute to run first in the list.

postscripts (postscripts.postscripts)

Comma separated list of scripts that should be run on this node after diskfull installation or diskless boot. Each script can take zero or more parameters. For example: "script1 p1 p2,script2,...". 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 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 installation of AIX, the scripts will run after the reboot and acts the same as the postbootscripts attribute. For AIX, use the postbootscripts attribute. Support will be added in the future for the postscripts attribute to run the scripts before the reboot in AIX.

power (nodehm.power)

The method to use to control the power of the node. If not set, the mgt attribute will be used. Valid values: ipmi, blade, hmc, ivm, fsp, kvm, esx, rhevm. If "ipmi", xCAT will search for this node in the ipmi table for more info. If "blade", xCAT will search for this node in the mp table. If "hmc", "ivm", or "fsp", xCAT will search for this node in the ppc table.

pprofile (ppc.pprofile)

The LPAR profile that will be used the next time the LPAR is powered on with rpower. For DFM, the pprofile attribute should be set to blank

prescripts-begin (prescripts.begin)

The scripts to be run at the beginning of the nodeset(Linux),

```
nimnodeset(AIX) or mkdsklsnode(AIX) command.

The format is:

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

where:

 - action1 and action2 for Linux are the nodeset actions specified in the
↪command.

   For AIX, action1 and action1 can be 'diskless' for mkdsklsnode command'

   and 'standalone for nimnodeset command.

 - 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  (all actions)

  diskless:myscript1,myscript2   (AIX)
```

```
  install:myscript1,myscript2|netboot:myscript3    (Linux)


All the scripts should be copied to /install/prescripts directory.

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

  NODES a coma separated list of node names that need to run the script for

  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

will be invoked at at a time. If it is not specified, the script will be␣
→invoked

once for all the nodes.
```

prescripts-end (prescripts.end)

>   The scripts to be run at the end of the nodeset(Linux),

```
nimnodeset(AIX),or mkdsklsnode(AIX) command.

The format is the same as the 'begin' column.
```

primarynic (noderes.primarynic)

>   This attribute will be deprecated. All the used network interface will be determined by installnic. The
>   network adapter on the node that will be used for xCAT management, the primarynic can be set to the
>   network adapter name or the mac address or the keyword "mac" which means that the network interface
>   specified by the mac address in the mac table will be used. Default is eth0.

productkey (prodkey.key)

>   The product key relevant to the aforementioned node/group and product combination

profile (nodetype.profile)

>   The string to use to locate a kickstart or autoyast template to use for OS deployment of this node. If the
>   provmethod attribute is set to an osimage name, that takes precedence, and profile need not be defined.
>   Otherwise, the os, profile, and arch are used to search for the files in /install/custom first, and then in
>   /opt/xcat/share/xcat.

provmethod (nodetype.provmethod)

>   The provisioning method for node deployment. The valid values are install, netboot, statelite or an os
>   image name from the osimage table. If an image name is specified, the osimage definition stored in the
>   osimage table and the linuximage table (for Linux) or nimimage table (for AIX) are used to locate the files
>   for templates, pkglists, syncfiles, etc. On Linux, if install, netboot or statelite is specified, the os, profile,
>   and arch are used to search for the files in /install/custom first, and then in /opt/xcat/share/xcat.

rack (nodepos.rack)

The frame the node is in.

room (nodepos.room)

The room where the node is located.

routenames (noderes.routenames)

A comma separated list of route names that refer to rows in the routes table. These are the routes that should be defined on this node when it is deployed.

serial (vpd.serial)

The serial number of the node.

serialflow (nodehm.serialflow)

The flow control value of the serial port for this node. For SOL this is typically 'hard'.

serialport (nodehm.serialport)

The serial port for this node, in the linux numbering style (0=COM1/ttyS0, 1=COM2/ttyS1). For SOL on IBM blades, this is typically 1. For rackmount IBM servers, this is typically 0.

serialspeed (nodehm.serialspeed)

The speed of the serial port for this node. For SOL this is typically 19200.

servicenode (noderes.servicenode)

A comma separated list of node names (as known by the management node) that provides most services for this node. The first service node on the list that is accessible will be used. The 2nd node on the list is generally considered to be the backup service node for this node when running commands like snmove.

setupconserver (servicenode.conserver)

Do we set up Conserver on this service node? Valid values:yes or 1, no or 0. If yes, configures and starts conserver daemon. If no or 0, it does not change the current state of the service.

setupdhcp (servicenode.dhcpserver)

Do we set up DHCP on this service node? Not supported on AIX. Valid values:yes or 1, no or 0. If yes, runs makedhcp -n. If no or 0, it does not change the current state of the service.

setupftp (servicenode.ftpserver)

Do we set up a ftp server on this service node? Not supported on AIX Valid values:yes or 1, no or 0. If yes, configure and start vsftpd. (You must manually install vsftpd on the service nodes before this.) If no or 0, it does not change the current state of the service. xCAT is not using ftp for compute nodes provisioning or any other xCAT features, so this attribute can be set to 0 if the ftp service will not be used for other purposes

setupipforward (servicenode.ipforward)

Do we set up ip forwarding on this service node? Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

setupldap (servicenode.ldapserver)

Do we set up ldap caching proxy on this service node? Not supported on AIX. Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

setupnameserver (servicenode.nameserver)

Do we set up DNS on this service node? Valid values: 2, 1, no or 0. If 2, creates named.conf as dns slave, using the management node as dns master, and starts named. If 1, creates named.conf file with forwarding to the management node and starts named. If no or 0, it does not change the current state of the service.

setupnfs (servicenode.nfsserver)

> Do we set up file services (HTTP,FTP,or NFS) on this service node? For AIX will only setup NFS, not HTTP or FTP. Valid values:yes or 1, no or 0.If no or 0, it does not change the current state of the service.

setupnim (servicenode.nimserver)

> Not used. Do we set up a NIM server on this service node? Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

setupntp (servicenode.ntpserver)

> Not used. Use setupntp postscript to setup a ntp server on this service node? Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

setupproxydhcp (servicenode.proxydhcp)

> Do we set up proxydhcp service on this node? valid values: yes or 1, no or 0. If yes, the proxydhcp daemon will be enabled on this node.

setuptftp (servicenode.tftpserver)

> Do we set up TFTP on this service node? Not supported on AIX. Valid values:yes or 1, no or 0. If yes, configures and starts atftp. If no or 0, it does not change the current state of the service.

sfp (ppc.sfp)

> The Service Focal Point of this Frame. This is the name of the HMC that is responsible for collecting hardware service events for this frame and all of the CECs within this frame.

side (vpd.side)

> <BPA>-<port> or <FSP>-<port>. The side information for the BPA/FSP. The side attribute refers to which BPA/FSP, A or B, which is determined by the slot value returned from lsslp command. It also lists the physical port within each BPA/FSP which is determined by the IP address order from the lsslp response. This information is used internally when communicating with the BPAs/FSPs

slot (nodepos.slot)

> The slot number of the blade in the chassis. For PCM, a comma-separated list of slot numbers is stored

slotid (mp.id)

> The slot number of this blade in the BladeCenter chassis.

slots (mpa.slots)

> The number of available slots in the chassis. For PCM, this attribute is used to store the number of slots in the following format: <slot rows>,<slot columns>,<slot orientation> Where:

```
<slot rows>  = number of rows of slots in chassis

<slot columns> = number of columns of slots in chassis

<slot orientation> = set to 0 if slots are vertical, and set to 1 if slots
↪of horizontal
```

storagcontroller (storage.controller)

> The management address to attach/detach new volumes.

```
In the scenario involving multiple controllers, this data must be

passed as argument rather than by table value
```

storagetype (storage.type)

> The plugin used to drive storage configuration (e.g. svc)

supernode (ppc.supernode)

> Indicates the connectivity of this CEC in the HFI network. A comma separated list of 2 ids. The first one is the supernode number the CEC is part of. The second one is the logical location number (0-3) of this CEC within the supernode.

supportedarchs (nodetype.supportedarchs)

> Comma delimited list of architectures this node can execute.

supportproxydhcp (noderes.proxydhcp)

> To specify whether the node supports proxydhcp protocol. Valid values: yes or 1, no or 0. Default value is yes.

switch (switch.switch)

> The switch hostname.

switchinterface (switch.interface)

> The interface name from the node perspective. For example, eth0. For the primary nic, it can be empty, the word "primary" or "primary:ethx" where ethx is the interface name.

switchport (switch.port)

> The port number in the switch that this node is connected to. On a simple 1U switch, an administrator can generally enter the number as printed next to the ports, and xCAT will understand switch representation differences. On stacked switches or switches with line cards, administrators should usually use the CLI representation (i.e. 2/0/1 or 5/8). One notable exception is stacked SMC 8848M switches, in which you must add 56 for the proceeding switch, then the port number. For example, port 3 on the second switch in an SMC8848M stack would be 59

switchvlan (switch.vlan)

> The ID for the tagged vlan that is created on this port using mkvlan and chvlan commands.

termport (nodehm.termport)

> The port number on the terminal server that this node is connected to.

termserver (nodehm.termserver)

> The hostname of the terminal server.

tftpdir (noderes.tftpdir)

> The directory that roots this nodes contents from a tftp and related perspective. Used for NAS offload by using different mountpoints.

tftpserver (noderes.tftpserver)

> The TFTP server for this node (as known by this node). If not set, it defaults to networks.tftpserver.

unit (nodepos.u)

> The vertical position of the node in the frame

urlpath (mpa.urlpath)

> URL path for the Chassis web interface. The full URL is built as follows: <hostname>/<urlpath>

usercomment (nodegroup.comments)

Any user-written notes.

userid (zvm.userid)

The z/VM userID of this node.

username (ppchcp.username, mpa.username, websrv.username)

Userid of the HMC or IVM. If not filled in, xCAT will look in the passwd table for key=hmc or key=ivm. If not in the passwd table, the default used is hscroot for HMCs and padmin for IVMs.

or

Userid to use to access the management module.

or

Userid to use to access the web service.

vmbeacon (vm.beacon)

This flag is used by xCAT to track the state of the identify LED with respect to the VM.

vmbootorder (vm.bootorder)

Boot sequence (i.e. net,hd)

vmcfgstore (vm.cfgstore)

Optional location for persistant storage separate of emulated hard drives for virtualization solutions that require persistant store to place configuration data

vmcluster (vm.cluster)

Specify to the underlying virtualization infrastructure a cluster membership for the hypervisor.

vmcpus (vm.cpus)

Number of CPUs the node should see.

vmhost (vm.host)

The system that currently hosts the VM

vmmanager (vm.mgr)

The function manager for the virtual machine

vmmaster (vm.master)

The name of a master image, if any, this virtual machine is linked to. This is generally set by clonevm and indicates the deletion of a master that would invalidate the storage of this virtual machine

vmmemory (vm.memory)

Megabytes of memory the VM currently should be set to.

vmnicnicmodel (vm.nicmodel)

Model of NICs that will be provided to VMs (i.e. e1000, rtl8139, virtio, etc)

vmnics (vm.nics)

Network configuration parameters. Of the general form [physnet:]interface,.. Generally, interface describes the vlan entity (default for native, tagged for tagged, vl[number] for a specific vlan. physnet is a virtual switch name or port description that is used for some virtualization technologies to construct virtual switches. hypervisor.netmap can map names to hypervisor specific layouts, or the descriptions described there may be used directly here where possible.

vmothersetting (vm.othersettings)

This allows specifying a semicolon delimited list of key->value pairs to include in a vmx file of VMware. For partitioning on normal power machines, this option is used to specify the hugepage and/or bsr information, the value is like:'hugepage:1,bsr=2'.

vmphyslots (vm.physlots)

Specify the physical slots drc index that will assigned to the partition, the delimiter is ',', and the drc index must started with '0x'. For more details, please reference to manpage of 'lsvm'.

vmstorage (vm.storage)

A list of storage files or devices to be used. i.e. dir:///cluster/vm/<nodename> or nfs://<server>/path/to/folder/

vmstoragecache (vm.storagecache)

Select caching scheme to employ. E.g. KVM understands 'none', 'writethrough' and 'writeback'

vmstorageformat (vm.storageformat)

Select disk format to use by default (e.g. raw versus qcow2)

vmstoragemodel (vm.storagemodel)

Model of storage devices to provide to guest

vmtextconsole (vm.textconsole)

Tracks the Psuedo-TTY that maps to the serial port or console of a VM

vmvirtflags (vm.virtflags)

General flags used by the virtualization method. For example, in Xen it could, among other things, specify paravirtualized setup, or direct kernel boot. For a hypervisor/dom0 entry, it is the virtualization method (i.e. "xen"). For KVM, the following flag=value pairs are recognized:

```
imageformat=[raw|fullraw|qcow2]

    raw is a generic sparse file that allocates storage on demand

    fullraw is a generic, non-sparse file that preallocates all space

    qcow2 is a sparse, copy-on-write capable format implemented at the
→virtualization layer rather than the filesystem level

clonemethod=[qemu-img|reflink]

    qemu-img allows use of qcow2 to generate virtualization layer copy-on-
→write

    reflink uses a generic filesystem facility to clone the files on your
→behalf, but requires filesystem support such as btrfs

placement_affinity=[migratable|user_migratable|pinned]
```

vmvncport (vm.vncport)

Tracks the current VNC display port (currently not meant to be set

webport (websrv.port)

The port of the web service.

wherevals (nodegroup.wherevals)

> A list of "attr*val" pairs that can be used to determine the members of a dynamic group, the delimiter is "::" and the operator * can be ==, =~, != or !~.

xcatmaster (noderes.xcatmaster)

> The hostname of the xCAT service node (as known by this node). This acts as the default value for nfsserver and tftpserver, if they are not set. If xcatmaster is not set, the node will use whoever responds to its boot request as its master. For the directed bootp case for POWER, it will use the management node if xcatmaster is not set.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

kit.7

NAME

kit - a logical object definition in the xCAT database.

SYNOPSIS

kit Attributes: *basename*, *description*, *isinternal*, *kitdeployparams*, *kitdir*, *kitname*, *ostype*, *release*, *version*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

kit Attributes:

basename (kit.basename)

> The kit base name

description (kit.description)

> The Kit description.

isinternal (kit.isinternal)

> A flag to indicated if the Kit is internally used. When set to 1, the Kit is internal. If 0 or undefined, the kit is not internal.

kitdeployparams (kit.kitdeployparams)

> The file containing the default deployment parameters for this Kit. These parameters are added to the OS Image definition.s list of deployment parameters when one or more Kit Components from this Kit are added to the OS Image.

kitdir (kit.kitdir)

> The path to Kit Installation directory on the Mgt Node.

kitname (kit.kitname)

> The unique generated kit name, when kit is added to the cluster.

ostype (kit.ostype)

> The kit OS type. Linux or AIX.

release (kit.release)

> The kit release

version (kit.version)

> The kit version

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

kitcomponent.7

NAME

kitcomponent - a logical object definition in the xCAT database.

SYNOPSIS

kitcomponent Attributes: *basename*, *description*, *driverpacks*, *exlist*, *genimage_postinstall*, *kitcompdeps*, *kitcompname*, *kitname*, *kitpkgdeps*, *kitreponame*, *postbootscripts*, *prerequisite*, *release*, *serverroles*, *version*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

kitcomponent Attributes:

basename (kitcomponent.basename)

> Kit Component basename.

description (kitcomponent.description)

> The Kit component description.

driverpacks (kitcomponent.driverpacks)

> Comma-separated List of driver package names. These must be full names like: pkg1-1.0-1.x86_64.rpm.

exlist (kitcomponent.exlist)

> Exclude list file containing the files/directories to exclude when building a diskless image.

genimage_postinstall (kitcomponent.genimage_postinstall)

> Comma-separated list of postinstall scripts that will run during the genimage.

kitcompdeps (kitcomponent.kitcompdeps)

> Comma-separated list of kit components that this kit component depends on.

kitcompname (kitcomponent.kitcompname)

> The unique Kit Component name. It is auto-generated when the parent Kit is added to the cluster.

kitname (kitcomponent.kitname)

> The Kit name which this Kit Component belongs to.

kitpkgdeps (kitcomponent.kitpkgdeps)

> Comma-separated list of packages that this kit component depends on.

kitreponame (kitcomponent.kitreponame)

> The Kit Package Repository name which this Kit Component belongs to.

postbootscripts (kitcomponent.postbootscripts)

> Comma-separated list of postbootscripts that will run during the node boot.

prerequisite (kitcomponent.prerequisite)

> Prerequisite for this kit component, the prerequisite includes ospkgdeps,preinstall,preupgrade,preuninstall scripts

release (kitcomponent.release)

> Kit Component release.

serverroles (kitcomponent.serverroles)

> The types of servers that this Kit Component can install on. Valid types are: mgtnode, servicenode, compute

version (kitcomponent.version)

> Kit Component version.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

kitrepo.7

NAME

kitrepo - a logical object definition in the xCAT database.

SYNOPSIS

kitrepo Attributes: *compat_osbasenames*, *kitname*, *kitrepodir*, *kitreponame*, *osarch*, *osbasename*, *osmajorversion*, *osminorversion*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

kitrepo Attributes:

compat_osbasenames (kitrepo.compat_osbasenames)

> List of compatible OS base names.

kitname (kitrepo.kitname)

> The Kit name which this Kit Package Repository belongs to.

kitrepodir (kitrepo.kitrepodir)

> The path to Kit Repository directory on the Mgt Node.

kitreponame (kitrepo.kitreponame)

> The unique generated kit repo package name, when kit is added to the cluster.

osarch (kitrepo.osarch)

> The OS distro arch which this repository is based on.

osbasename (kitrepo.osbasename)

> The OS distro name which this repository is based on.

osmajorversion (kitrepo.osmajorversion)

> The OS distro major version which this repository is based on.

osminorversion (kitrepo.osminorversion)

> The OS distro minor version which this repository is based on. If this attribute is not set, it means that this repo applies to all minor versions.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

monitoring.7

NAME

monitoring - a logical object definition in the xCAT database.

SYNOPSIS

monitoring Attributes: *comments*, *disable*, *name*, *nodestatmon*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

monitoring Attributes:

comments (monitoring.comments)

> Any user-written notes.

disable (monitoring.disable)

> Set to 'yes' or '1' to comment out this row.

name (monitoring.name)

> The name of the mornitoring plug-in module. The plug-in must be put in /lib/perl/xCAT_monitoring/. See the man page for monstart for details.

nodestatmon (monitoring.nodestatmon)

> Specifies if the monitoring plug-in is used to feed the node status to the xCAT cluster. Any one of the following values indicates "yes": y, Y, yes, Yes, YES, 1. Any other value or blank (default), indicates "no".

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

network.7

NAME

network - a logical object definition in the xCAT database.

SYNOPSIS

network Attributes: *ddnsdomain*, *dhcpserver*, *domain*, *dynamicrange*, *gateway*, *logservers*, *mask*, *mgtifname*, *nameservers*, *net*, *netname*, *nodehostname*, *ntpservers*, *staticrange*, *staticrangeincrement*, *tftpserver*, *usercomment*, *vlanid*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

network Attributes:

ddnsdomain (networks.ddnsdomain)

> A domain to be combined with nodename to construct FQDN for DDNS updates induced by DHCP. This is not passed down to the client as "domain"

dhcpserver (networks.dhcpserver)

> The DHCP server that is servicing this network. Required to be explicitly set for pooled service node operation.

domain (networks.domain)

> The DNS domain name (ex. cluster.com).

dynamicrange (networks.dynamicrange)

> The IP address range used by DHCP to assign dynamic IP addresses for requests on this network. This should not overlap with entities expected to be configured with static host declarations, i.e. anything ever expected to be a node with an address registered in the mac table.

gateway (networks.gateway)

> The network gateway. It can be set to an ip address or the keyword <xcatmaster>, the keyword <xcatmaster> indicates the cluster-facing ip address configured on this management node or service node. Leaving this field blank means that there is no gateway for this network.

logservers (networks.logservers)

> The log servers for this network. Used in creating the DHCP network definition. Assumed to be the DHCP server if not set.

mask (networks.mask)

> The network mask.

mgtifname (networks.mgtifname)

> The interface name of the management/service node facing this network. !remote!<nicname> indicates a non-local network on a specific nic for relay DHCP.

nameservers (networks.nameservers)

> A comma delimited list of DNS servers that each node in this network should use. This value will end up in the nameserver settings of the /etc/resolv.conf on each node in this network. If this attribute value is set to the IP address of an xCAT node, make sure DNS is running on it. In a hierarchical cluster, you can also set this attribute to "<xcatmaster>" to mean the DNS server for each node in this network should be the node that is managing it (either its service node or the management node). Used in creating the DHCP network definition, and DNS configuration.

net (networks.net)

> The network address.

netname (networks.netname)

Name used to identify this network definition.

nodehostname (networks.nodehostname)

A regular expression used to specify node name to network-specific hostname. i.e. "/z/-secondary/" would mean that the hostname of "n1" would be n1-secondary on this network. By default, the nodename is assumed to equal the hostname, followed by nodename-interfacename.

ntpservers (networks.ntpservers)

The ntp servers for this network. Used in creating the DHCP network definition. Assumed to be the DHCP server if not set.

staticrange (networks.staticrange)

The IP address range used to dynamically assign static IPs to newly discovered nodes. This should not overlap with the dynamicrange nor overlap with entities that were manually assigned static IPs. The format for the attribute value is: <startip>-<endip>.

staticrangeincrement (networks.staticrangeincrement)

tftpserver (networks.tftpserver)

The TFTP server that is servicing this network. If not set, the DHCP server is assumed.

usercomment (networks.comments)

Any user-written notes.

vlanid (networks.vlanid)

The vlan ID if this network is within a vlan.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

node.7

NAME

node - a logical object definition in the xCAT database.

SYNOPSIS

node Attributes: *addkcmdline, appstatus, appstatustime, arch, authdomain, bmc, bmcpassword, bmcport, bmcusername, cfgmgr, cfgmgtroles, cfgserver, chain, chassis, cmdmapping, cons, conserver, consoleondemand, cpucount, cputype, currchain, currstate, dhcpinterfaces, disksize, displayname, domainadminpassword, domainadminuser, domaintype, getmac, groups, hcp, height, hidden, hostcluster, hostinterface, hostmanager, hostnames, hosttype, hwtype, id, initrd, installnic, interface, ip, iscsipassword, iscsiserver, iscsitarget, iscsiuserid, kcmdline, kernel, mac, memory, mgt, micbridge, michost, micid, miconboot, micpowermgt, micvlog, migrationdest, monserver, mpa, mtm, nameservers, netboot, nfsdir, nfsserver, nicaliases, niccustomscripts, nicextraparams, nichostnameprefixes, nichostnamesuffixes, nicips, nicnetworks, nictypes, nimserver, node, nodetype, ondiscover, os, osvolume, otherinterfaces, ou, parent, passwd.HMC, passwd.admin, passwd.celogin, passwd.general, passwd.hscroot, password, postbootscripts, postscripts, power, pprofile, prescripts-begin, prescripts-end, primarynic, primarysn, productkey, profile, provmethod, rack, room, routenames, serial, serialflow, serialport, serialspeed, servicenode, setupconserver, setupdhcp, setupftp, setupipforward, setupldap, setupnameserver, setupnfs, setupnim, setupntp, setupproxydhcp, setuptftp, sfp, side, slot,*

slotid, slots, status, statustime, storagcontroller, storagetype, supernode, supportedarchs, supportproxydhcp, switch, switchinterface, switchport, switchvlan, termport, termserver, tftpdir, tftpserver, unit, updatestatus, updatestatustime, urlpath, usercomment, userid, username, vmbeacon, vmbootorder, vmcfgstore, vmcluster, vmcpus, vmhost, vmmanager, vmmaster, vmmemory, vmnicnicmodel, vmnics, vmothersetting, vmphyslots, vmstorage, vmstoragecache, vmstorageformat, vmstoragemodel, vmtextconsole, vmvirtflags, vmvncport, webport, xcatmaster, zonename

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

node Attributes:

addkcmdline (bootparams.addkcmdline)

> User specified one or more parameters to be passed to the kernel. For the kernel options need to be persistent after installation, specify them with prefix "R::"

appstatus (nodelist.appstatus)

> A comma-delimited list of application status. For example: 'sshd=up,ftp=down,ll=down'

appstatustime (nodelist.appstatustime)

> The date and time when appstatus was updated.

arch (nodetype.arch)

> The hardware architecture of this node. Valid values: x86_64, ppc64, x86, ia64.

authdomain (domain.authdomain)

> If a node should participate in an AD domain or Kerberos realm distinct from domain indicated in site, this field can be used to specify that

bmc (ipmi.bmc)

> The hostname of the BMC adapater.

bmcpassword (ipmi.password)

> The BMC password. If not specified, the key=ipmi row in the passwd table is used as the default.

bmcport (ipmi.bmcport)

```
In systems with selectable shared/dedicated ethernet ports,

        this parameter can be used to specify the preferred port.  0

        means use the shared port, 1 means dedicated, blank is to not

        assign.


        The following special cases exist for IBM System x servers:
```

```
        For x3755 M3 systems, 0 means use the dedicated port, 1 means

        shared, blank is to not assign.


    For certain systems which have a mezzaine or ML2 adapter, there is a␣
↪second

    value to include:



        For x3750 M4 (Model 8722):



        0 2   1st 1Gbps interface for LOM


        0 0   1st 10Gbps interface for LOM


        0 3   2nd 1Gbps interface for LOM


        0 1   2nd 10Gbps interface for LOM




        For  x3750 M4 (Model 8752), x3850/3950 X6, dx360 M4, x3550 M4, and␣
↪x3650 M4:




        0     Shared (1st onboard interface)


        1     Dedicated


        2 0   First interface on ML2 or mezzanine adapter
```

```
          2 1    Second interface on ML2 or mezzanine adapter



          2 2    Third interface on ML2 or mezzanine adapter



          2 3    Fourth interface on ML2 or mezzanine adapter
```

bmcusername (ipmi.username)

> The BMC userid. If not specified, the key=ipmi row in the passwd table is used as the default.

cfgmgr (cfgmgt.cfgmgr)

> The name of the configuration manager service. Currently 'chef' and 'puppet' are supported services.

cfgmgtroles (cfgmgt.roles)

> The roles associated with this node as recognized by the cfgmgr for the software that is to be installed and configured. These role names map to chef recipes or puppet manifest classes that should be used for this node. For example, chef OpenStack cookbooks have roles such as mysql-master,keystone, glance, nova-controller, nova-conductor, cinder-all.

cfgserver (cfgmgt.cfgserver)

> The xCAT node name of the chef server or puppet master

chain (chain.chain)

> A comma-delimited chain of actions to be performed automatically when this node is discovered. ("Discovered" means a node booted, but xCAT and DHCP did not recognize the MAC of this node. In this situation, xCAT initiates the discovery process, the last step of which is to run the operations listed in this chain attribute, one by one.) Valid values: boot or reboot, install or netboot, runcmd=<cmd>, runimage=<URL>, shell, standby. (Default - same as no chain - it will do only the discovery.). Example, for BMC machines use: runcmd=bmcsetup,shell.

chassis (nodepos.chassis)

> The BladeCenter chassis the blade is in.

cmdmapping (nodehm.cmdmapping)

> The fully qualified name of the file that stores the mapping between PCM hardware management commands and xCAT/third-party hardware management commands for a particular type of hardware device. Only used by PCM.

cons (nodehm.cons)

> The console method. If nodehm.serialport is set, this will default to the nodehm.mgt setting, otherwise it defaults to unused. Valid values: cyclades, mrv, or the values valid for the mgt attribute.

conserver (nodehm.conserver)

> The hostname of the machine where the conserver daemon is running. If not set, the default is the xCAT management node.

consoleondemand (nodehm.consoleondemand)

> This overrides the value from site.consoleondemand; (0=no, 1=yes). Default is the result from site.consoleondemand.

cpucount (hwinv.cpucount)

> The number of cpus for the node.

cputype (hwinv.cputype)

> The cpu model name for the node.

currchain (chain.currchain)

> The chain steps still left to do for this node. This attribute will be automatically adjusted by xCAT while xCAT-genesis is running on the node (either during node discovery or a special operation like firmware update). During node discovery, this attribute is initialized from the chain attribute and updated as the chain steps are executed.

currstate (chain.currstate)

> The current or next chain step to be executed on this node by xCAT-genesis. Set by xCAT during node discovery or as a result of nodeset.

dhcpinterfaces (servicenode.dhcpinterfaces)

> The network interfaces DHCP server should listen on for the target node. This attribute can be used for management node and service nodes. If defined, it will override the values defined in site.dhcpinterfaces. This is a comma separated list of device names. !remote! indicates a non-local network for relay DHCP. For example: !remote!,eth0,eth1

disksize (hwinv.disksize)

> The size of the disks for the node in GB.

displayname (mpa.displayname)

> Alternative name for BladeCenter chassis. Only used by PCM.

domainadminpassword (domain.adminpassword)

> Allow a node specific indication of Administrative user password for the domain. Most will want to ignore this in favor of passwd table.

domainadminuser (domain.adminuser)

> Allow a node specific indication of Administrative user. Most will want to just use passwd table to indicate this once rather than by node.

domaintype (domain.type)

> Type, if any, of authentication domain to manipulate. The only recognized value at the moment is activedirectory.

getmac (nodehm.getmac)

> The method to use to get MAC address of the node with the getmac command. If not set, the mgt attribute will be used. Valid values: same as values for mgmt attribute.

groups (nodelist.groups)

> A comma-delimited list of groups this node is a member of. Group names are arbitrary, except all nodes should be part of the 'all' group. Internal group names are designated by using __<groupname>. For example, __Unmanaged, could be the internal name for a group of nodes that is not managed by xCAT. Admins should avoid using the __ characters when defining their groups.

hcp (ppc.hcp, zvm.hcp)

> The hardware control point for this node (HMC, IVM, Frame or CEC). Do not need to set for BPAs and FSPs.

or

The hardware control point for this node.

height (nodepos.height)

The server height in U(s).

hidden (nodelist.hidden)

Used to hide fsp and bpa definitions, 1 means not show them when running lsdef and nodels

hostcluster (hypervisor.cluster)

Specify to the underlying virtualization infrastructure a cluster membership for the hypervisor.

hostinterface (hypervisor.interface)

The definition of interfaces for the hypervisor. The format is [network-name:interfacename:bootprotocol:IP:netmask:gateway] that split with | for each interface

hostmanager (hypervisor.mgr)

The virtualization specific manager of this hypervisor when applicable

hostnames (hosts.hostnames)

Hostname aliases added to /etc/hosts for this node. Comma or blank separated list.

hosttype (hypervisor.type)

The plugin associated with hypervisor specific commands such as revacuate

hwtype (ppc.nodetype, zvm.nodetype, mp.nodetype, mic.nodetype)

The hardware type of the node. Only can be one of fsp, bpa, cec, frame, ivm, hmc and lpar

or

The node type. Valid values: cec (Central Electronic Complex), lpar (logical partition), zvm (z/VM host operating system), and vm (virtual machine).

or

The hardware type for mp node. Valid values: mm,cmm, blade.

or

The hardware type of the mic node. Generally, it is mic.

id (ppc.id, mp.id)

For LPARs: the LPAR numeric id; for CECs: the cage number; for Frames: the frame number.

or

The slot number of this blade in the BladeCenter chassis.

initrd (bootparams.initrd)

The initial ramdisk image that network boot actions should use (could be a DOS floppy or hard drive image if using memdisk as kernel)

installnic (noderes.installnic)

The network adapter on the node that will be used for OS deployment, the installnic can be set to the network adapter name or the mac address or the keyword "mac" which means that the network interface specified by the mac address in the mac table will be used. If not set, primarynic will be used. If primarynic is not set too, the keyword "mac" will be used as default.

---

interface (mac.interface)

> The adapter interface name that will be used to install and manage the node. E.g. eth0 (for linux) or en0 (for AIX).)

ip (hosts.ip)

> The IP address of the node. This is only used in makehosts. The rest of xCAT uses system name resolution to resolve node names to IP addresses.

iscsipassword (iscsi.passwd)

> The password for the iscsi server containing the boot device for this node.

iscsiserver (iscsi.server)

> The server containing the iscsi boot device for this node.

iscsitarget (iscsi.target)

> The iscsi disk used for the boot device for this node. Filled in by xCAT.

iscsiuserid (iscsi.userid)

> The userid of the iscsi server containing the boot device for this node.

kcmdline (bootparams.kcmdline)

> Arguments to be passed to the kernel

kernel (bootparams.kernel)

> The kernel that network boot actions should currently acquire and use. Note this could be a chained boot loader such as memdisk or a non-linux boot loader

mac (mac.mac)

> The mac address or addresses for which xCAT will manage static bindings for this node. This may be simply a mac address, which would be bound to the node name (such as "01:02:03:04:05:0E"). This may also be a "|" delimited string of "mac address!hostname" format (such as "01:02:03:04:05:0E!node5|01:02:03:05:0F!node6-eth1").

memory (hwinv.memory)

> The size of the memory for the node in MB.

mgt (nodehm.mgt)

> The method to use to do general hardware management of the node. This attribute is used as the default if power or getmac is not set. Valid values: ipmi, blade, hmc, ivm, fsp, bpa, kvm, esx, rhevm. See the power attribute for more details.

micbridge (mic.bridge)

> The virtual bridge on the host node which the mic connected to.

michost (mic.host)

> The host node which the mic card installed on.

micid (mic.id)

> The device id of the mic node.

miconboot (mic.onboot)

> Set mic to autoboot when mpss start. Valid values: yes|no. Default is yes.

micpowermgt (mic.powermgt)

Set the Power Management for mic node. This attribute is used to set the power management state that mic may get into when it is idle. Four states can be set: cpufreq, corec6, pc3 and pc6. The valid value for powermgt attribute should be [cpufreq=<on|off>]![corec6=<on|off>]![pc3=<on|off>]![pc6=<on|off>]. e.g. cpufreq=on!corec6=off!pc3=on!pc6=off. Refer to the doc of mic to get more information for power management.

micvlog (mic.vlog)

Set the Verbose Log to console. Valid values: yes|no. Default is no.

migrationdest (vm.migrationdest)

A noderange representing candidate destinations for migration (i.e. similar systems, same SAN, or other criteria that xCAT can use

monserver (noderes.monserver)

The monitoring aggregation point for this node. The format is "x,y" where x is the ip address as known by the management node and y is the ip address as known by the node.

mpa (mp.mpa)

The managment module used to control this blade.

mtm (vpd.mtm)

The machine type and model number of the node. E.g. 7984-6BU

nameservers (noderes.nameservers)

An optional node/group specific override for name server list. Most people want to stick to site or network defined nameserver configuration.

netboot (noderes.netboot)

The type of network booting to use for this node. Valid values: pxe or xnba for x86* architecture, yaboot for POWER architecture, grub2-tftp and grub2-http for RHEL7 on Power and all the os deployment on Power LE. Notice: yaboot is not supported from rhels7 on Power,use grub2-tftp or grub2-http instead, the difference between the 2 is the file transfer protocol(i.e, http or tftp)

nfsdir (noderes.nfsdir)

The path that should be mounted from the NFS server.

nfsserver (noderes.nfsserver)

The NFS or HTTP server for this node (as known by this node).

nicaliases (nics.nicaliases)

Comma-separated list of hostname aliases for each NIC.

```
Format: eth0!<alias list>,eth1!<alias1 list>|<alias2 list>

        For multiple aliases per nic use a space-separated list.

For example: eth0!moe larry curly,eth1!tom|jerry
```

niccustomscripts (nics.niccustomscripts)

Comma-separated list of custom scripts per NIC. <nic1>!<script1>,<nic2>!<script2>, e.g. eth0!configeth eth0, ib0!configib ib0. The xCAT object definition commands support to use niccustomscripts.<nicname> as the sub attribute

.

---

nicextraparams (nics.nicextraparams)

Comma-separated list of extra parameters that will be used for each NIC configuration.

```
    If only one ip address is associated with each NIC:

        <nic1>!<param1=value1 param2=value2>,<nic2>!<param3=value3>, for␣
→example, eth0!MTU=1500,ib0!MTU=65520 CONNECTED_MODE=yes.

    If multiple ip addresses are associated with each NIC:

        <nic1>!<param1=value1 param2=value2>|<param3=value3>,<nic2>!
→<param4=value4 param5=value5>|<param6=value6>, for example, eth0!
→MTU=1500|MTU=1460,ib0!MTU=65520 CONNECTED_MODE=yes.

The xCAT object definition commands support to use nicextraparams.<nicname>␣
→as the sub attributes.
```

nichostnameprefixes (nics.nichostnameprefixes)

Comma-separated list of hostname prefixes per NIC.

```
If only one ip address is associated with each NIC:

    <nic1>!<ext1>,<nic2>!<ext2>,..., for example, eth0!eth0-,ib0!ib-

If multiple ip addresses are associated with each NIC:

    <nic1>!<ext1>|<ext2>,<nic2>!<ext1>|<ext2>,..., for example,  eth0!eth0-
→|eth0-ipv6i-,ib0!ib-|ib-ipv6-.

The xCAT object definition commands support to use nichostnameprefixes.
→<nicname> as the sub attributes.

Note:  According to DNS rules a hostname must be a text string up to 24␣
→characters drawn from the alphabet (A-Z), digits (0-9), minus sign (-),and␣
→period (.). When you are specifying "nichostnameprefixes" or "nicaliases"␣
→make sure the resulting hostnames will conform to this naming convention
```

nichostnamesuffixes (nics.nichostnamesuffixes)

Comma-separated list of hostname suffixes per NIC.

```
If only one ip address is associated with each NIC:

    <nic1>!<ext1>,<nic2>!<ext2>,..., for example, eth0!-eth0,ib0!-ib0

If multiple ip addresses are associated with each NIC:

    <nic1>!<ext1>|<ext2>,<nic2>!<ext1>|<ext2>,..., for example,  eth0!-eth0|-
→eth0-ipv6,ib0!-ib0|-ib0-ipv6.

The xCAT object definition commands support to use nichostnamesuffixes.
→<nicname> as the sub attributes.

Note:  According to DNS rules a hostname must be a text string up to 24␣
→characters drawn from the alphabet (A-Z), digits (0-9), minus sign (-),and␣
→period (.). When you are specifying "nichostnamesuffixes" or "nicaliases"␣
→make sure the resulting hostnames will conform to this naming convention
```

nicips (nics.nicips)

Comma-separated list of IP addresses per NIC. To specify one ip address per NIC:

```
    <nic1>!<ip1>,<nic2>!<ip2>,..., for example, eth0!10.0.0.100,ib0!11.0.0.
→100

To specify multiple ip addresses per NIC:

    <nic1>!<ip1>|<ip2>,<nic2>!<ip1>|<ip2>,..., for example, eth0!10.0.0.
→100|fd55::214:5eff:fe15:849b,ib0!11.0.0.100|2001::214:5eff:fe15:849a. The␣
→xCAT object definition commands support to use nicips.<nicname> as the sub␣
→attributes.

Note: The primary IP address must also be stored in the hosts.ip attribute.␣
→The nichostnamesuffixes should specify one hostname suffix for each ip␣
→address.
```

nicnetworks (nics.nicnetworks)

Comma-separated list of networks connected to each NIC.

```
If only one ip address is associated with each NIC:

    <nic1>!<network1>,<nic2>!<network2>, for example, eth0!10_0_0_0-255_255_
→0_0, ib0!11_0_0_0-255_255_0_0

If multiple ip addresses are associated with each NIC:

    <nic1>!<network1>|<network2>,<nic2>!<network1>|<network2>, for example,␣
→eth0!10_0_0_0-255_255_0_0|fd55:faaf:e1ab:336::/64,ib0!11_0_0_0-255_255_0_
→0|2001:db8:1:0::/64. The xCAT object definition commands support to use␣
→nicnetworks.<nicname> as the sub attributes.
```

nictypes (nics.nictypes)

Comma-separated list of NIC types per NIC. <nic1>!<type1>,<nic2>!<type2>, e.g. eth0!Ethernet,ib0!Infiniband. The xCAT object definition commands support to use nictypes.<nicname> as the sub attributes.

nimserver (noderes.nimserver)

Not used for now. The NIM server for this node (as known by this node).

node (nodelist.node)

The hostname of a node in the cluster.

nodetype (nodetype.nodetype)

A comma-delimited list of characteristics of this node. Valid values: ppc, blade, vm (virtual machine), osi (OS image), mm, mn, rsa, switch.

ondiscover (chain.ondiscover)

This attribute is currently not used by xCAT. The "nodediscover" operation is always done during node discovery.

os (nodetype.os)

The operating system deployed on this node. Valid values: AIX, rhels*,rhelc*, rhas*,centos*,SL*, fedora*, sles* (where * is the version #). As a special case, if this is set to "boottarget", then it will use

the initrd/kernel/parameters specified in the row in the boottarget table in which boottarget.bprofile equals nodetype.profile.

osvolume (storage.osvolume)

Specification of what storage to place the node OS image onto. Examples include:

```
localdisk (Install to first non-FC attached disk)

usbdisk (Install to first USB mass storage device seen)

wwn=0x50000393c813840c (Install to storage device with given WWN)
```

otherinterfaces (hosts.otherinterfaces)

Other IP addresses to add for this node. Format: -<ext>:<ip>,<intfhostname>:<ip>,...

ou (domain.ou)

For an LDAP described machine account (i.e. Active Directory), the orginaztional unit to place the system. If not set, defaults to cn=Computers,dc=your,dc=domain

parent (ppc.parent)

For LPARs: the CEC; for FSPs: the CEC; for CEC: the frame (if one exists); for BPA: the frame; for frame: the building block number (which consists 1 or more service nodes and compute/storage nodes that are serviced by them - optional).

passwd.HMC (ppcdirect.password)

Password of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

passwd.admin (ppcdirect.password)

Password of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

passwd.celogin (ppcdirect.password)

Password of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

passwd.general (ppcdirect.password)

Password of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

passwd.hscroot (ppcdirect.password)

Password of the FSP/BPA(for ASMI) and CEC/Frame(for DFM). If not filled in, xCAT will look in the passwd table for key=fsp. If not in the passwd table, the default used is admin.

password (ppchcp.password, mpa.password, websrv.password)

Password of the HMC or IVM. If not filled in, xCAT will look in the passwd table for key=hmc or key=ivm. If not in the passwd table, the default used is abc123 for HMCs and padmin for IVMs.

or

Password to use to access the management module. If not specified, the key=blade row in the passwd table is used as the default.

or

Password to use to access the web service.

postbootscripts (postscripts.postbootscripts)

Comma separated list of scripts that should be run on this node after diskfull installation or diskless boot. Each script can take zero or more parameters. For example: "script1 p1 p2,script2,...". On AIX these scripts are run during the processing of /etc/inittab. On Linux they are run at the init.d time. xCAT automatically adds the scripts in the xcatdefaults.postbootscripts attribute to run first in the list.

postscripts (postscripts.postscripts)

Comma separated list of scripts that should be run on this node after diskfull installation or diskless boot. Each script can take zero or more parameters. For example: "script1 p1 p2,script2,...". 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 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 installation of AIX, the scripts will run after the reboot and acts the same as the postbootscripts attribute. For AIX, use the postbootscripts attribute. Support will be added in the future for the postscripts attribute to run the scripts before the reboot in AIX.

power (nodehm.power)

The method to use to control the power of the node. If not set, the mgt attribute will be used. Valid values: ipmi, blade, hmc, ivm, fsp, kvm, esx, rhevm. If "ipmi", xCAT will search for this node in the ipmi table for more info. If "blade", xCAT will search for this node in the mp table. If "hmc", "ivm", or "fsp", xCAT will search for this node in the ppc table.

pprofile (ppc.pprofile)

The LPAR profile that will be used the next time the LPAR is powered on with rpower. For DFM, the pprofile attribute should be set to blank

prescripts-begin (prescripts.begin)

The scripts to be run at the beginning of the nodeset(Linux),

```
nimnodeset(AIX) or mkdsklsnode(AIX) command.

The format is:

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

where:
 - action1 and action2 for Linux are the nodeset actions specified in the␣
↪command.

    For AIX, action1 and action1 can be 'diskless' for mkdsklsnode command'

    and 'standalone for nimnodeset command.

 - 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  (all actions)
```

```
  diskless:myscript1,myscript2    (AIX)

  install:myscript1,myscript2|netboot:myscript3    (Linux)


All the scripts should be copied to /install/prescripts directory.

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

  NODES a coma separated list of node names that need to run the script for

  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

will be invoked at at a time. If it is not specified, the script will be␣
→invoked

once for all the nodes.
```

prescripts-end (prescripts.end)

  The scripts to be run at the end of the nodeset(Linux),

```
nimnodeset(AIX),or mkdsklsnode(AIX) command.

The format is the same as the 'begin' column.
```

primarynic (noderes.primarynic)

  This attribute will be deprecated. All the used network interface will be determined by installnic. The network adapter on the node that will be used for xCAT management, the primarynic can be set to the network adapter name or the mac address or the keyword "mac" which means that the network interface specified by the mac address in the mac table will be used. Default is eth0.

primarysn (nodelist.primarysn)

  Not used currently. The primary servicenode, used by this node.

productkey (prodkey.key)

  The product key relevant to the aforementioned node/group and product combination

profile (nodetype.profile)

  The string to use to locate a kickstart or autoyast template to use for OS deployment of this node. If the provmethod attribute is set to an osimage name, that takes precedence, and profile need not be defined. Otherwise, the os, profile, and arch are used to search for the files in /install/custom first, and then in /opt/xcat/share/xcat.

provmethod (nodetype.provmethod)

  The provisioning method for node deployment. The valid values are install, netboot, statelite or an os image name from the osimage table. If an image name is specified, the osimage definition stored in the

osimage table and the linuximage table (for Linux) or nimimage table (for AIX) are used to locate the files for templates, pkglists, syncfiles, etc. On Linux, if install, netboot or statelite is specified, the os, profile, and arch are used to search for the files in /install/custom first, and then in /opt/xcat/share/xcat.

rack (nodepos.rack)

The frame the node is in.

room (nodepos.room)

The room where the node is located.

routenames (noderes.routenames)

A comma separated list of route names that refer to rows in the routes table. These are the routes that should be defined on this node when it is deployed.

serial (vpd.serial)

The serial number of the node.

serialflow (nodehm.serialflow)

The flow control value of the serial port for this node. For SOL this is typically 'hard'.

serialport (nodehm.serialport)

The serial port for this node, in the linux numbering style (0=COM1/ttyS0, 1=COM2/ttyS1). For SOL on IBM blades, this is typically 1. For rackmount IBM servers, this is typically 0.

serialspeed (nodehm.serialspeed)

The speed of the serial port for this node. For SOL this is typically 19200.

servicenode (noderes.servicenode)

A comma separated list of node names (as known by the management node) that provides most services for this node. The first service node on the list that is accessible will be used. The 2nd node on the list is generally considered to be the backup service node for this node when running commands like snmove.

setupconserver (servicenode.conserver)

Do we set up Conserver on this service node? Valid values:yes or 1, no or 0. If yes, configures and starts conserver daemon. If no or 0, it does not change the current state of the service.

setupdhcp (servicenode.dhcpserver)

Do we set up DHCP on this service node? Not supported on AIX. Valid values:yes or 1, no or 0. If yes, runs makedhcp -n. If no or 0, it does not change the current state of the service.

setupftp (servicenode.ftpserver)

Do we set up a ftp server on this service node? Not supported on AIX Valid values:yes or 1, no or 0. If yes, configure and start vsftpd. (You must manually install vsftpd on the service nodes before this.) If no or 0, it does not change the current state of the service. xCAT is not using ftp for compute nodes provisioning or any other xCAT features, so this attribute can be set to 0 if the ftp service will not be used for other purposes

setupipforward (servicenode.ipforward)

Do we set up ip forwarding on this service node? Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

setupldap (servicenode.ldapserver)

Do we set up ldap caching proxy on this service node? Not supported on AIX. Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

setupnameserver (servicenode.nameserver)

>   Do we set up DNS on this service node? Valid values: 2, 1, no or 0. If 2, creates named.conf as dns slave, using the management node as dns master, and starts named. If 1, creates named.conf file with forwarding to the management node and starts named. If no or 0, it does not change the current state of the service.

setupnfs (servicenode.nfsserver)

>   Do we set up file services (HTTP,FTP,or NFS) on this service node? For AIX will only setup NFS, not HTTP or FTP. Valid values:yes or 1, no or 0.If no or 0, it does not change the current state of the service.

setupnim (servicenode.nimserver)

>   Not used. Do we set up a NIM server on this service node? Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

setupntp (servicenode.ntpserver)

>   Not used. Use setupntp postscript to setup a ntp server on this service node? Valid values:yes or 1, no or 0. If no or 0, it does not change the current state of the service.

setupproxydhcp (servicenode.proxydhcp)

>   Do we set up proxydhcp service on this node? valid values: yes or 1, no or 0. If yes, the proxydhcp daemon will be enabled on this node.

setuptftp (servicenode.tftpserver)

>   Do we set up TFTP on this service node? Not supported on AIX. Valid values:yes or 1, no or 0. If yes, configures and starts atftp. If no or 0, it does not change the current state of the service.

sfp (ppc.sfp)

>   The Service Focal Point of this Frame. This is the name of the HMC that is responsible for collecting hardware service events for this frame and all of the CECs within this frame.

side (vpd.side)

>   <BPA>-<port> or <FSP>-<port>. The side information for the BPA/FSP. The side attribute refers to which BPA/FSP, A or B, which is determined by the slot value returned from lsslp command. It also lists the physical port within each BPA/FSP which is determined by the IP address order from the lsslp response. This information is used internally when communicating with the BPAs/FSPs

slot (nodepos.slot)

>   The slot number of the blade in the chassis. For PCM, a comma-separated list of slot numbers is stored

slotid (mp.id)

>   The slot number of this blade in the BladeCenter chassis.

slots (mpa.slots)

>   The number of available slots in the chassis. For PCM, this attribute is used to store the number of slots in the following format: <slot rows>,<slot columns>,<slot orientation> Where:

```
<slot rows>  = number of rows of slots in chassis

<slot columns> = number of columns of slots in chassis

<slot orientation> = set to 0 if slots are vertical, and set to 1 if slots
→of horizontal
```

status (nodelist.status)

The current status of this node. This attribute will be set by xCAT software. Valid values: defined, booting, netbooting, booted, discovering, configuring, installing, alive, standingby, powering-off, unreachable. If blank, defined is assumed. The possible status change sequenses are: For installaton: defined->[discovering]->[configuring]->[standingby]->installing->booting->booted->[alive], For diskless deployment: defined->[discovering]->[configuring]->[standingby]->netbooting->booted->[alive], For booting: [alive/unreachable]->booting->[alive], For powering off: [alive]->powering-off->[unreachable], For monitoring: alive->unreachable. Discovering and configuring are for x Series dicovery process. Alive and unreachable are set only when there is a monitoring plug-in start monitor the node status for xCAT. Please note that the status values will not reflect the real node status if you change the state of the node from outside of xCAT (i.e. power off the node using HMC GUI).

statustime (nodelist.statustime)

The data and time when the status was updated.

storagcontroller (storage.controller)

The management address to attach/detach new volumes.

```
In the scenario involving multiple controllers, this data must be

passed as argument rather than by table value
```

storagetype (storage.type)

The plugin used to drive storage configuration (e.g. svc)

supernode (ppc.supernode)

Indicates the connectivity of this CEC in the HFI network. A comma separated list of 2 ids. The first one is the supernode number the CEC is part of. The second one is the logical location number (0-3) of this CEC within the supernode.

supportedarchs (nodetype.supportedarchs)

Comma delimited list of architectures this node can execute.

supportproxydhcp (noderes.proxydhcp)

To specify whether the node supports proxydhcp protocol. Valid values: yes or 1, no or 0. Default value is yes.

switch (switch.switch)

The switch hostname.

switchinterface (switch.interface)

The interface name from the node perspective. For example, eth0. For the primary nic, it can be empty, the word "primary" or "primary:ethx" where ethx is the interface name.

switchport (switch.port)

The port number in the switch that this node is connected to. On a simple 1U switch, an administrator can generally enter the number as printed next to the ports, and xCAT will understand switch representation differences. On stacked switches or switches with line cards, administrators should usually use the CLI representation (i.e. 2/0/1 or 5/8). One notable exception is stacked SMC 8848M switches, in which you must add 56 for the proceeding switch, then the port number. For example, port 3 on the second switch in an SMC8848M stack would be 59

switchvlan (switch.vlan)

The ID for the tagged vlan that is created on this port using mkvlan and chvlan commands.

termport (nodehm.termport)

> The port number on the terminal server that this node is connected to.

termserver (nodehm.termserver)

> The hostname of the terminal server.

tftpdir (noderes.tftpdir)

> The directory that roots this nodes contents from a tftp and related perspective. Used for NAS offload by using different mountpoints.

tftpserver (noderes.tftpserver)

> The TFTP server for this node (as known by this node). If not set, it defaults to networks.tftpserver.

unit (nodepos.u)

> The vertical position of the node in the frame

updatestatus (nodelist.updatestatus)

> The current node update status. Valid states are synced, out-of-sync,syncing,failed.

updatestatustime (nodelist.updatestatustime)

> The date and time when the updatestatus was updated.

urlpath (mpa.urlpath)

> URL path for the Chassis web interface. The full URL is built as follows: <hostname>/<urlpath>

usercomment (nodelist.comments)

> Any user-written notes.

userid (zvm.userid)

> The z/VM userID of this node.

username (ppchcp.username, mpa.username, websrv.username)

> Userid of the HMC or IVM. If not filled in, xCAT will look in the passwd table for key=hmc or key=ivm. If not in the passwd table, the default used is hscroot for HMCs and padmin for IVMs.
>
> or
>
> Userid to use to access the management module.
>
> or
>
> Userid to use to access the web service.

vmbeacon (vm.beacon)

> This flag is used by xCAT to track the state of the identify LED with respect to the VM.

vmbootorder (vm.bootorder)

> Boot sequence (i.e. net,hd)

vmcfgstore (vm.cfgstore)

> Optional location for persistant storage separate of emulated hard drives for virtualization solutions that require persistant store to place configuration data

vmcluster (vm.cluster)

> Specify to the underlying virtualization infrastructure a cluster membership for the hypervisor.

---

vmcpus (vm.cpus)

>   Number of CPUs the node should see.

vmhost (vm.host)

>   The system that currently hosts the VM

vmmanager (vm.mgr)

>   The function manager for the virtual machine

vmmaster (vm.master)

>   The name of a master image, if any, this virtual machine is linked to. This is generally set by clonevm and indicates the deletion of a master that would invalidate the storage of this virtual machine

vmmemory (vm.memory)

>   Megabytes of memory the VM currently should be set to.

vmnicnicmodel (vm.nicmodel)

>   Model of NICs that will be provided to VMs (i.e. e1000, rtl8139, virtio, etc)

vmnics (vm.nics)

>   Network configuration parameters. Of the general form [physnet:]interface,.. Generally, interface describes the vlan entity (default for native, tagged for tagged, vl[number] for a specific vlan. physnet is a virtual switch name or port description that is used for some virtualization technologies to construct virtual switches. hypervisor.netmap can map names to hypervisor specific layouts, or the descriptions described there may be used directly here where possible.

vmothersetting (vm.othersettings)

>   This allows specifying a semicolon delimited list of key->value pairs to include in a vmx file of VMware. For partitioning on normal power machines, this option is used to specify the hugepage and/or bsr information, the value is like:'hugepage:1,bsr=2'.

vmphyslots (vm.physlots)

>   Specify the physical slots drc index that will assigned to the partition, the delimiter is ',', and the drc index must started with '0x'. For more details, please reference to manpage of 'lsvm'.

vmstorage (vm.storage)

>   A list of storage files or devices to be used. i.e. dir:///cluster/vm/<nodename> or nfs://<server>/path/to/folder/

vmstoragecache (vm.storagecache)

>   Select caching scheme to employ. E.g. KVM understands 'none', 'writethrough' and 'writeback'

vmstorageformat (vm.storageformat)

>   Select disk format to use by default (e.g. raw versus qcow2)

vmstoragemodel (vm.storagemodel)

>   Model of storage devices to provide to guest

vmtextconsole (vm.textconsole)

>   Tracks the Psuedo-TTY that maps to the serial port or console of a VM

vmvirtflags (vm.virtflags)

General flags used by the virtualization method. For example, in Xen it could, among other things, specify paravirtualized setup, or direct kernel boot. For a hypervisor/dom0 entry, it is the virtualization method (i.e. "xen"). For KVM, the following flag=value pairs are recognized:

```
imageformat=[raw|fullraw|qcow2]

    raw is a generic sparse file that allocates storage on demand

    fullraw is a generic, non-sparse file that preallocates all space

    qcow2 is a sparse, copy-on-write capable format implemented at the␣
→virtualization layer rather than the filesystem level

clonemethod=[qemu-img|reflink]

    qemu-img allows use of qcow2 to generate virtualization layer copy-on-
→write

    reflink uses a generic filesystem facility to clone the files on your␣
→behalf, but requires filesystem support such as btrfs

placement_affinity=[migratable|user_migratable|pinned]
```

vmvncport (vm.vncport)

Tracks the current VNC display port (currently not meant to be set

webport (websrv.port)

The port of the web service.

xcatmaster (noderes.xcatmaster)

The hostname of the xCAT service node (as known by this node). This acts as the default value for nfsserver and tftpserver, if they are not set. If xcatmaster is not set, the node will use whoever responds to its boot request as its master. For the directed bootp case for POWER, it will use the management node if xcatmaster is not set.

zonename (nodelist.zonename)

The name of the zone to which the node is currently assigned. If undefined, then it is not assigned to any zone.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

notification.7

NAME

notification - a logical object definition in the xCAT database.

SYNOPSIS

notification Attributes: *comments*, *filename*, *tableops*, *tables*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

notification Attributes:

comments (notification.comments)

>   Any user-written notes.

filename (notification.filename)

>   The path name of a file that implements the callback routine when the monitored table changes. Can be a perl module or a command. See the regnotif man page for details.

tableops (notification.tableops)

>   Specifies the table operation to monitor for. Valid values: "d" (rows deleted), "a" (rows added), "u" (rows updated).

tables (notification.tables)

>   Comma-separated list of xCAT database tables to monitor.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

osdistro.7

NAME

osdistro - a logical object definition in the xCAT database.

SYNOPSIS

osdistro Attributes: *arch*, *basename*, *dirpaths*, *majorversion*, *minorversion*, *osdistroname*, *type*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

osdistro Attributes:

arch (osdistro.arch)

> The OS distro arch (e.g. x86_64)

basename (osdistro.basename)

> The OS base name (e.g. rhels)

dirpaths (osdistro.dirpaths)

> Directory paths where OS distro is store. There could be multiple paths if OS distro has more than one ISO image. (e.g. /install/rhels6.2/x86_64,...)

majorversion (osdistro.majorversion)

> The OS distro major version.(e.g. 6)

minorversion (osdistro.minorversion)

> The OS distro minor version. (e.g. 2)

osdistroname (osdistro.osdistroname)

> Unique name (e.g. rhels6.2-x86_64)

type (osdistro.type)

> Linux or AIX

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

osdistroupdate.7

NAME

osdistroupdate - a logical object definition in the xCAT database.

SYNOPSIS

osdistroupdate Attributes: *dirpath*, *downloadtime*, *osdistroname*, *osupdatename*, *usercomment*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

osdistroupdate Attributes:

dirpath (osdistroupdate.dirpath)

> Path to where OS distro update is stored. (e.g. /install/osdistroupdates/rhels6.2-x86_64-20120716-update)

downloadtime (osdistroupdate.downloadtime)

> The timestamp when OS distro update was downloaded..

osdistroname (osdistroupdate.osdistroname)

> The OS distro name to update. (e.g. rhels)

osupdatename (osdistroupdate.osupdatename)

> Name of OS update. (e.g. rhn-update1)

usercomment (osdistroupdate.comments)

> Any user-written notes.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

osimage.7

NAME

osimage - a logical object definition in the xCAT database.

SYNOPSIS

osimage Attributes: *addkcmdline, boottarget, bosinst_data, cfmdir, configdump, crashkernelsize, description, driverupdatesrc, dump, exlist, fb_script, groups, home, image_data, imagename, imagetype, installp_bundle, installto, isdeletable, kerneldir, kernelver, kitcomponents, krpmver, lpp_source, mksysb, netdrivers, nimmethod, nimtype, nodebootif, osarch, osdistroname, osname, osupdatename, osvers, otherifce, otherpkgdir, otherpkglist, otherpkgs, paging, partitionfile, permission, pkgdir, pkglist, postbootscripts, postinstall, postscripts, profile, provmethod, resolv_conf, root, rootfstype, rootimgdir, script, serverrole, shared_home, shared_root, spot, synclists, template, tmp, usercomment, winpepath*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

osimage Attributes:

addkcmdline (linuximage.addkcmdline)

> User specified arguments to be passed to the kernel. The user arguments are appended to xCAT.s default kernel arguments. For the kernel options need to be persistent after installation, specify them with prefix "R::". This attribute is ignored if linuximage.boottarget is set.

boottarget (linuximage.boottarget)

> The name of the boottarget definition. When this attribute is set, xCAT will use the kernel, initrd and kernel params defined in the boottarget definition instead of the default.

bosinst_data (nimimage.bosinst_data)

> The name of a NIM bosinst_data resource.

cfmdir (osimage.cfmdir)

> CFM directory name for PCM. Set to /install/osimages/<osimage name>/cfmdir by PCM.

configdump (nimimage.configdump)

> Specifies the type of system dump to be collected. The values are selective, full, and none. The default is selective.

crashkernelsize (linuximage.crashkernelsize)

> the size that assigned to the kdump kernel. If the kernel size is not set, 256M will be the default value.

description (osimage.description)

> OS Image Description

driverupdatesrc (linuximage.driverupdatesrc)

> The source of the drivers which need to be loaded during the boot. Two types of driver update source are supported: Driver update disk and Driver rpm package. The value for this attribute should be comma separated sources. Each source should be the format tab:full_path_of_srouce_file. The tab keyword can be: dud (for Driver update disk) and rpm (for driver rpm). If missing the tab, the rpm format is the default. e.g. dud:/install/dud/dd.img,rpm:/install/rpm/d.rpm

dump (linuximage.dump, nimimage.dump)

> The NFS directory to hold the Linux kernel dump file (vmcore) when the node with this image crashes, its format is "nfs://<nfs_server_ip>/<kdump_path>". If you want to use the node's "xcatmaster" (its SN or MN), <nfs_server_ip> can be left blank. For example, "nfs:///<kdump_path>" means the NFS directory to hold the kernel dump file is on the node's SN, or MN if there's no SN.

> or

> The name of the NIM dump resource.

exlist (linuximage.exlist)

> The fully qualified name of the file that stores the file names and directory names that will be excluded from the image during packimage command. It is used for diskless image only.

fb_script (nimimage.fb_script)

> The name of a NIM fb_script resource.

groups (osimage.groups)

> A comma-delimited list of image groups of which this image is a member. Image groups can be used in the litefile and litetree table instead of a single image name. Group names are arbitrary.

home (nimimage.home)

> The name of the NIM home resource.

image_data (nimimage.image_data)

> The name of a NIM image_data resource.

imagename (osimage.imagename)

> The name of this xCAT OS image definition.

imagetype (osimage.imagetype)

> The type of operating system image this definition represents (linux,AIX).

installp_bundle (nimimage.installp_bundle)

> One or more comma separated NIM installp_bundle resources.

installto (winimage.installto)

> The disk and partition that the Windows will be deployed to. The valid format is <disk>:<partition>. If not set, default value is 0:1 for bios boot mode(legacy) and 0:3 for uefi boot mode; If setting to 1, it means 1:1 for bios boot and 1:3 for uefi boot

isdeletable (osimage.isdeletable)

> A flag to indicate whether this image profile can be deleted. This attribute is only used by PCM.

kerneldir (linuximage.kerneldir)

> The directory name where the 3rd-party kernel is stored. It is used for diskless image only.

kernelver (linuximage.kernelver)

> The version of linux kernel used in the linux image. If the kernel version is not set, the default kernel in rootimgdir will be used

kitcomponents (osimage.kitcomponents)

> List of Kit Component IDs assigned to this OS Image definition.

krpmver (linuximage.krpmver)

> The rpm version of kernel packages (for SLES only). If it is not set, the default rpm version of kernel packages will be used.

lpp_source (nimimage.lpp_source)

> The name of the NIM lpp_source resource.

mksysb (nimimage.mksysb)

> The name of a NIM mksysb resource.

netdrivers (linuximage.netdrivers)

> The ethernet device drivers of the nodes which will use this linux image, at least the device driver for the nodes' installnic should be included

nimmethod (nimimage.nimmethod)

> The NIM install method to use, (ex. rte, mksysb).

nimtype (nimimage.nimtype)

> The NIM client type- standalone, diskless, or dataless.

nodebootif (linuximage.nodebootif)

The network interface the stateless/statelite node will boot over (e.g. eth0)

osarch (osimage.osarch)

The hardware architecture of this node. Valid values: x86_64, ppc64, x86, ia64.

osdistroname (osimage.osdistroname)

The name of the OS distro definition. This attribute can be used to specify which OS distro to use, instead of using the osname,osvers,and osarch attributes. For *kit commands, the attribute will be used to read the osdistro table for the osname, osvers, and osarch attributes. If defined, the osname, osvers, and osarch attributes defined in the osimage table will be ignored.

osname (osimage.osname)

Operating system name- AIX or Linux.

osupdatename (osimage.osupdatename)

A comma-separated list of OS distro updates to apply to this osimage.

osvers (osimage.osvers)

The Linux operating system deployed on this node. Valid values: rhels*,rhelc*, rhas*,centos*,SL*, fedora*, sles* (where * is the version #).

otherifce (linuximage.otherifce)

Other network interfaces (e.g. eth1) in the image that should be configured via DHCP

otherpkgdir (linuximage.otherpkgdir)

The base directory where the non-distro packages are stored. Only 1 local directory supported at present.

otherpkglist (linuximage.otherpkglist)

The fully qualified name of the file that stores non-distro package lists that will be included in the image. It could be set multiple paths.The multiple paths must be seperated by ",".

otherpkgs (nimimage.otherpkgs)

One or more comma separated installp or rpm packages. The rpm packages must have a prefix of 'R:', (ex. R:foo.rpm)

paging (nimimage.paging)

The name of the NIM paging resource.

partitionfile (linuximage.partitionfile, winimage.partitionfile)

The path of the configuration file which will be used to partition the disk for the node. For stateful osimages,two types of files are supported: "<partition file absolute path>" which contains a partitioning definition that will be inserted directly into the generated autoinst configuration file and must be formatted for the corresponding OS installer (e.g. kickstart for RedHat, autoyast for SLES, pressed for Ubuntu). "s:<partitioning script absolute path>" which specifies a shell script that will be run from the OS installer configuration file %pre section; the script must write the correct partitioning definition into the file /tmp/partitionfile on the node which will be included into the configuration file during the install process. For statelite osimages, partitionfile should specify "<partition file absolute path>"; see the xCAT Statelite documentation for the xCAT defined format of this configuration file.For Ubuntu, besides "<partition file absolute path>" or "s:<partitioning script absolute path>", the disk name(s) to partition must be specified in traditional, non-devfs format, delimited with space, it can be specified in 2 forms: "d:<the absolute path of the disk name file>" which contains the disk name(s) to partition and "s:d:<the absolute path of the disk script>" which runs in pressed/early_command and writes the disk names into the "/tmp/boot_disk" . To support other specific partition methods such as RAID or LVM in Ubuntu, some additional preseed values should be specified, these values can be specified with "c:<the absolute path of the additional pressed

config file>" which contains the additional pressed entries in "d-i ..." form and "s:c:<the absolute path of the additional pressed config script>" which runs in pressed/early_command and set the preseed values with "debconf-set". The multiple values should be delimited with comma ","

or

The path of partition configuration file. Since the partition configuration for bios boot mode and uefi boot mode are different, this configuration file can include both configurations if you need to support both bios and uefi mode. Either way, you must specify the boot mode in the configuration. Example of partition configuration file: [BIOS]xxxxxxx[UEFI]yyyyyyy. To simplify the setting, you also can set installto in partitionfile with section like [INSTALLTO]0:1

permission (linuximage.permission)

The mount permission of /.statelite directory is used, its default value is 755

pkgdir (linuximage.pkgdir)

The name of the directory where the distro packages are stored. It could be set multiple paths.The multiple paths must be seperated by ",". The first path in the value of osimage.pkgdir must be the OS base pkg dir path, such as pkgdir=/install/rhels6.2/x86_64,/install/updates . In the os base pkg path, there are default repository data. And in the other pkg path(s), the users should make sure there are repository data. If not, use "createrepo" command to create them. For ubuntu, multiple mirrors can be specified in the pkgdir attribute, the mirrors must be prefixed by the protocol(http/ssh) and delimited with "," between each other.

pkglist (linuximage.pkglist)

The fully qualified name of the file that stores the distro packages list that will be included in the image. Make sure that if the pkgs in the pkglist have dependency pkgs, the dependency pkgs should be found in one of the pkgdir

postbootscripts (osimage.postbootscripts)

Comma separated list of scripts that should be run on this after diskfull installation or diskless boot. On AIX these scripts are run during the processing of /etc/inittab. On Linux they are run at the init.d time. xCAT automatically adds the scripts in the xcatdefaults.postbootscripts attribute to run first in the list. See the site table runbootscripts attribute.

postinstall (linuximage.postinstall)

The fully qualified name of the script file that will be run at the end of the genimage command. It could be set multiple paths.The multiple paths must be seperated by ",". It is used for diskless image only.

postscripts (osimage.postscripts)

Comma separated list of scripts that should be run on this image after diskfull installation 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 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 installation of AIX, the scripts will run after the reboot and acts the same as the postbootscripts attribute. For AIX, use the postbootscripts attribute. See the site table runbootscripts attribute. Support will be added in the future for the postscripts attribute to run the scripts before the reboot in AIX.

profile (osimage.profile)

The node usage category. For example compute, service.

provmethod (osimage.provmethod)

The provisioning method for node deployment. The valid values are install, netboot,statelite,boottarget,dualboot,sysclone. If boottarget is set, you must set linuximage.boottarget to the name of the boottarget definition. It is not used by AIX.

resolv_conf (nimimage.resolv_conf)

> The name of the NIM resolv_conf resource.

root (nimimage.root)

> The name of the NIM root resource.

rootfstype (osimage.rootfstype)

> The filesystem type for the rootfs is used when the provmethod is statelite. The valid values are nfs or ramdisk. The default value is nfs

rootimgdir (linuximage.rootimgdir)

> The directory name where the image is stored. It is generally used for diskless image. it also can be used in sysclone environment to specify where the image captured from golden client is stored. in sysclone environment, rootimgdir is generally assigned to some default value by xcat, but you can specify your own store directory. just one thing need to be noticed, wherever you save the image, the name of last level directory must be the name of image. for example, if your image name is testimage and you want to save this image under home directoy, rootimgdir should be assigned to value /home/testimage/

script (nimimage.script)

> The name of a NIM script resource.

serverrole (osimage.serverrole)

> The role of the server created by this osimage. Default roles: mgtnode, servicenode, compute, login, storage, utility.

shared_home (nimimage.shared_home)

> The name of the NIM shared_home resource.

shared_root (nimimage.shared_root)

> A shared_root resource represents a directory that can be used as a / (root) directory by one or more diskless clients.

spot (nimimage.spot)

> The name of the NIM SPOT resource.

synclists (osimage.synclists)

> The fully qualified name of a file containing a list of files to synchronize on the nodes. Can be a comma separated list of multiple synclist files. The synclist generated by PCM named /install/osimages/<imagename>/synclist.cfm is reserved for use only by PCM and should not be edited by the admin.

template (linuximage.template, winimage.template)

> The fully qualified name of the template file that will be used to create the OS installer configuration file for stateful installations (e.g. kickstart for RedHat, autoyast for SLES).
>
> or
>
> The fully qualified name of the template file that is used to create the windows unattend.xml file for diskful installation.

tmp (nimimage.tmp)

> The name of the NIM tmp resource.

usercomment (linuximage.comments, nimimage.comments)

Any user-written notes.

or

Any user-provided notes.

winpepath (winimage.winpepath)

> The path of winpe which will be used to boot this image. If the real path is /tftpboot/winboot/winpe1/, the value for winpepath should be set to winboot/winpe1

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

policy.7

NAME

policy - a logical object definition in the xCAT database.

SYNOPSIS

policy Attributes: *commands*, *host*, *name*, *noderange*, *parameters*, *priority*, *rule*, *time*, *usercomment*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

policy Attributes:

commands (policy.commands)

> The list of commands that this rule applies to. Default is "*" (all commands).

host (policy.host)

> The host from which users may issue the commands specified by this rule. Default is "*" (all hosts). Only all or one host is supported

name (policy.name)

> The username that is allowed to perform the commands specified by this rule. Default is "*" (all users).

noderange (policy.noderange)

> The Noderange that this rule applies to. Default is "*" (all nodes). Not supported with the *def commands.

parameters (policy.parameters)

> A regular expression that matches the command parameters (everything except the noderange) that this rule applies to. Default is "*" (all parameters). Not supported with the *def commands.

priority (policy.priority)

>   The priority value for this rule. This value is used to identify this policy data object (i.e. this rule) The table is sorted on this field with the lower the number the higher the priority. For example 1.0 is higher priority than 4.1 is higher than 4.9.

rule (policy.rule)

>   Specifies how this rule should be applied. Valid values are: allow, accept, trusted. Allow or accept will allow the user to run the commands. Any other value will deny the user access to the commands. Trusted means that once this client has been authenticated via the certificate, all other information that is sent (e.g. the username) is believed without question. This authorization should only be given to the xcatd on the management node at this time.

time (policy.time)

>   Time ranges that this command may be executed in. This is not supported.

usercomment (policy.comments)

>   Any user-written notes.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

rack.7

NAME

rack - a logical object definition in the xCAT database.

SYNOPSIS

rack Attributes: *displayname*, *height*, *num*, *rackname*, *room*, *usercomment*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

rack Attributes:

displayname (rack.displayname)

>   Alternative name for rack. Only used by PCM.

height (rack.height)

>   Number of units which can be stored in the rack.

num (rack.num)

The rack number.

rackname (rack.rackname)

The rack name.

room (rack.room)

The room in which the rack is located.

usercomment (rack.comments)

Any user-written notes.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

route.7

NAME

route - a logical object definition in the xCAT database.

SYNOPSIS

route Attributes: *gateway*, *ifname*, *mask*, *net*, *routename*, *usercomment*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

route Attributes:

gateway (routes.gateway)

The gateway that routes the ip traffic from the mn to the nodes. It is usually a service node.

ifname (routes.ifname)

The interface name that facing the gateway. It is optional for IPv4 routes, but it is required for IPv6 routes.

mask (routes.mask)

The network mask.

net (routes.net)

The network address.

routename (routes.routename)

Name used to identify this route.

usercomment (routes.comments)

>   Any user-written notes.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

site.7

NAME

site - a logical object definition in the xCAT database.

SYNOPSIS

site Attributes: *installdir*, *master*, *xcatdport*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

site Attributes:

installdir (site.value)

>   The installation directory

master (site.value)

>   The management node

xcatdport (site.value)

>   Port used by xcatd daemon on master

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

zone.7

NAME

zone - a logical object definition in the xCAT database.

SYNOPSIS

zone Attributes: *defaultzone*, *sshbetweennodes*, *sshkeydir*, *usercomment*, *zonename*

DESCRIPTION

Logical objects of this type are stored in the xCAT database in one or more tables. Use the following commands to manipulate the objects: **mkdef**, **chdef**, **lsdef**, and **rmdef**. These commands will take care of knowing which tables the object attributes should be stored in. The attribute list below shows, in parentheses, what tables each attribute is stored in.

zone Attributes:

defaultzone (zone.defaultzone)

> If nodes are not assigned to any other zone, they will default to this zone. If value is set to yes or 1.

sshbetweennodes (zone.sshbetweennodes)

> Indicates whether passwordless ssh will be setup between the nodes of this zone. Values are yes/1 or no/0. Default is yes.

sshkeydir (zone.sshkeydir)

> Directory containing the shared root ssh RSA keys.

usercomment (zone.comments)

> Any user-provided notes.

zonename (zone.zonename)

> The name of the zone.

SEE ALSO

mkdef(1), **chdef(1)**, **lsdef(1)**, **rmdef(1)**

man8

chtab.8

NAME

chtab - Add, delete or update rows in the database tables.

SYNOPSIS

chtab [-h| –help]

chtab [-v| –version]

*chtab [keycolname=keyvalue] [tablename.colname=newvalue] *

*chtab [keycolname=keyvalue] [tablename.colname+=newvalue] *

*chtab -d [keycolname=keyvalue] [tablename.colname=newvalue] *

DESCRIPTION

The chtab command adds, deletes or updates the attribute value in the specified table.column for the specified keyvalue. Normally, the given value will completely replace the current attribute value. But if "+=" is used instead of "=", the specified value will be appended to the coma separated list of the attribute, if it is not already there.

OPTIONS

-h Display usage message.

-v Command Version.

-d Delete option.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To add a node=node1 to the nodelist table with groups=all:

*chtab node=node1 nodelist.groups=all *

2. To add a keyword (tftpdir) and value (/tftpboot) to the site table:

*chtab key=tftpdir site.value=/tftpboot *

3. To add node1 to the nodetype table with os=rhel5:

chtab node=node1 nodetype.os=rhel5

4. To change node1 in nodetype table setting os=sles:

chtab node=node1 nodetype.os=sles

5. To change node1 by appending otherpkgs to the postbootscripts field in the postscripts table:

chtab node=node1 postscripts.postbootscripts+=otherpkgs

6. To delete node1 from nodetype table:

chtab -d node=node1 nodetype

FILES

/opt/xcat/bin/chtab

SEE ALSO

tabdump(8)|tabdump.8, tabedit(8)|tabedit.8

copycds-cdrom.8

SYNOPSIS

copycds-cdrom *[copycds options] [drive]*

DESCRIPTION

copycds-cdrom is a wrapper scripts for copycds to copy from physical CD/DVD-ROM drives located on the management server.

[copycds options] are passed unmolested to copycds.

If *[drive]* is not specified, /dev/cdrom is assumed.

The copycds command copies all contents of Distribution CDs or Service Pack CDs to the install directory as designated in the **site** table attribute: **installdir**.

SEE ALSO

copycds(8)|copycds.8

AUTHOR

Isaac Freeman <ifreeman@us.ibm.com>

copycds.8

NAME

copycds - Copies Linux distributions and service levels from DVDs/ISOs to the xCAT /install directory.

SYNOPSIS

copycds [{**-n**|**--name**|**--osver**}=*distroname*] [{**-a**|**--arch**}=*architecture*] [{**-p**|**--path**}=*ospkgpath*] [**-o** | **--noosimage**] [**-w** | **--nonoverwrite**] {*iso*|*device-path*} ...

copycds [**-i** | **--inspection**] {*iso*|*device-path*}

copycds [**-h** | **--help**]

DESCRIPTION

The **copycds** command copies all contents of Distribution DVDs/ISOs or Service Pack DVDs/ISOs to a destination directory. The destination directory can be specified by the -p option. If no path is specified, the default destination directory will be formed from the **installdir** site attribute and the distro name and architecture, for example: /install/rhels6.3/x86_64. The **copycds** command can copy from one or more ISO files, or the CD/DVD device path.

You can specify -i or --inspection option to check whether the DVDs/ISOs can be recognized by xCAT. If recognized, the distribution name, architecture and the disc no (the disc sequence number of DVDs/ISOs in multi-disk distribution) of the DVD/ISO is displayed. If xCAT doesn't recognize the DVD/ISO, you must manually specify the distro name and architecture using the -n and -a options. This is sometimes the case for distros that have very recently been released, and the xCAT code hasn't been updated for it yet.

You can get xCAT to recognize new DVDs/ISOs by adding them to /opt/xcat/lib/perl/xCAT/data/discinfo.pm and reloading xcatd (service xcatd reload).

OPTIONS

{**-n|--name|--osver**}=*distroname*

> The linux distro name and version that the ISO/DVD contains. Examples: rhels6.3, sles11.2, fedora9. Note the 's' in rhels6.3 which denotes the Server version of RHEL, which is typically used.

{**-a|--arch**}=*architecture*

> The architecture of the linux distro on the ISO/DVD. Examples: x86, x86_64, ppc64, s390x.

{**-p|--path**}=*ospkgpath*

> The destination directory to which the contents of ISO/DVD will be copied. When this option is not specified, the default destination directory will be formed from the **installdir** site attribute and the distro name and architecture, for example: /install/rhel6.3/x86_64. This option is only supported distributions of sles,redhat and windows.

{**-i|--inspection**}

> Check whether xCAT can recognize the DVDs/ISOs in the argument list, but do not copy the disc. Displays the os distribution name, architecture and disc no of each recognized DVD/ISO. This option only supported for distributions of sles,redhat and windows.

{**-o|--noosimage**}

> Do not create the default osimages based on the osdistro copied in. By default, copycds will create a set of osimages based on the osdistro.

{**-w|--nonoverwrite**}

> Complain and exit if the os disc has already been copied in. By default, copycds will overwrite the os disc already copied in.

RETURN VALUE

0: The command completed successfully. For the --inspection option, the ISO/DVD have been recognized successfully

Nonzero: An Error has occurred. For the --inspection option, the ISO/DVD cannot be recognized

EXAMPLES

*

To copy the RPMs from a set of ISOs that represent the DVDs of a distro:

```
copycds dvd1.iso dvd2.iso
```

*

To copy the RPMs from a physical DVD to /depot/kits/3 directory:

```
copycds -p /depot/kits/3 /dev/dvd
```

*

To copy the RPMs from a DVD ISO of a very recently released distro:

```
copycds -n rhels6.4 -a x86_64 dvd.iso
```

*

To check whether a DVD ISO can be recognized by xCAT and display the recognized disc info:

```
copycds -i /media/RHEL/6.2/RHEL6.2-20111117.0-Server-ppc64-DVD1.iso
```

Output will be similar to:

```
OS Image:/media/RHEL/6.2/RHEL6.2-20111117.0-Server-ppc64-DVD1.iso
DISTNAME:rhels6.2
ARCH:ppc64
DISCNO:1
```

For the attributes that are not recognized, the value will be blank.

*

To copy the packages from a supplemental DVD ISO file:

```
copycds -n /isodir/RHEL6.5/RHEL6.5-Supplementary-20131114.2-Server-ppc64-
↪DVD1.iso -n rhels6.5-supp
```

Also, remember to add the new directory to your osimage definition:

```
chdef -t osimage myosimage -p pkgdir=/install/rhels6.5-supp/ppc64
```

SEE ALSO

nodeset(8)|nodeset.8, site(5)|site.5, nodetype(5)|nodetype.5

makeconservercf.8

NAME

makeconservercf - creates the conserver configuration file from info in the xCAT database

SYNOPSIS

makeconservercf [**-V|--verbose**] [**-d|--delete**] *noderange*

makeconservercf [**-V|--verbose**] [**-l|--local**] [*noderange*]

makeconservercf [**-V|--verbose**] [**-c|--conserver**] [*noderange*]

makeconservercf [**-V|--verbose**] *noderange* [**-t|--trust**] *hosts*

makeconservercf [**-h|--help|-v|--version**]

DESCRIPTION

The **makeconservercf** command will write out the /etc/conserver.cf, using information from the nodehm table and related tables (e.g. mp, ipmi, ppc). Normally, **makeconservercf** will write all nodes to the conserver.cf file. If a noderange is specified, it will write on those nodes to the file. In either case, if a node does not have nodehm.cons set, it will not be written to the file.

In the case of a hierarchical cluster (i.e. one with service nodes) **makeconservercf** will determine which nodes will have their consoles accessed from the management node and which from a service node (based on the nodehm.conserver attribute). The /etc/conserver.cf file will be created accordingly on all relevant management/service nodes. If -l is specified, it will only create the local file.

OPTIONS

-d|--delete

Delete rather than add or refresh the nodes specified as a noderange.

-c|--conserver

Only set up the conserver on the conserver host. If no conserver host is set for nodes, the conserver gets set up only on the management node.

-l|--local

Only run **makeconservercf** locally and create the local /etc/conserver.cf. The default is to also run it on all service nodes, if there are any.

-t|--trust *hosts*

Add additional trusted hosts into /etc/conserver.cf. The *hosts* are comma separated list of ip addresses or host names.

-v|--version

Display version.

-V|--verbose

Verbose mode.

-h|--help

Display usage message.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To create conserver configuration for all the nodes.

```
makeconservercf
```

2. To create conserver configuration for nodes node01-node10.

```
makeconservercf node01-node10
```

3. To remove conserver configuration for node01.

```
makeconservercf -d node01
```

SEE ALSO

rcons(1)|rcons.1

makedhcp.8

NAME

makedhcp - Creates and updates DHCP configuration files.

SYNOPSIS

makedhcp -n [**-l** | **--localonly**]

makedhcp -a [**-l** | **--localonly**]

makedhcp -a -d [**-l** | **--localonly**]

makedhcp -d *noderange* [**-l** | **--localonly**]

makedhcp *noderange* [**-s** *statements*] [**-l** | **--localonly**]

makedhcp -q *noderange*

makedhcp [*-h*|-***-help*]

DESCRIPTION

The **makedhcp** command creates and updates the DHCP configuration on the management node and service nodes. The **makedhcp** command is supported for both Linux and AIX clusters.

1. Start by filling out the networks(5)|networks.5 table properly.

---

2. Then use the **makedhcp -n** option to create a new dhcp configuration file. You can set the site table, dhcplease attribute to the lease time for the dhcp client. The default value is 43200.

3. Next, get the node IP addresses and MACs defined in the xCAT database. Also, get the hostnames and IP addresses pushed to /etc/hosts (using makehosts(8)|makehosts.8) and to DNS (using makedns(8)|makedns.8).

4. Then run **makedhcp** with a noderange or the **-a** option. This will inject into dhcpd configuration data pertinent to the specified nodes. On linux, the configuration information immediately takes effect without a restart of DHCP.

If you need to delete node entries from the DHCP configuration, use the **-d** flag.

OPTIONS

-n

> Create a new dhcp configuration file with a network statement for each network the dhcp daemon should listen on. (Which networks dhcpd should listen on can be controlled by the dhcpinterfaces attribute in the site(5)|site.5 table.) The **makedhcp** command will automatically restart the dhcp daemon after this operation. This option will replace any existing configuration file (making a backup of it first). For Linux systems the file will include network entries as well as certain general parameters such as a dynamic range and omapi configuration. For AIX systems the file will include network entries. On AIX systems, if there are any non-xCAT entries in the existing configuration file they will be preserved and added to the end of the new configuration file.

-a

> Define all nodes to the DHCP server. (Will only add nodes that can be reached, network-wise, by this DHCP server.) The dhcp daemon does not have to be restarted after this. On AIX systems **makedhcp** will not add entries for cluster nodes that will be installed using NIM. The entries for these nodes will be managed by NIM.

noderange

> Add the specified nodes to the DHCP server configuration.

-s *statements*

> For the input noderange, the argument will be interpreted like dhcp configuration file text.

-d *noderange*

> Delete node entries from the DHCP server configuration. On AIX, any entries created by NIM will not be removed.

-a -d

> Delete all node entries, that were added by xCAT, from the DHCP server configuration.

-l | --localonly

> Configure dhcpd on the local machine only. Without this option, makedhcp will also send this operation to any service nodes that service the nodes in the noderange.

-q *noderange*

> Query the node entries from the DHCP server configuration. On AIX, any entries created by NIM will not be listed.

-h|--help

> Display usage message.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. Create a new DHCP configuration file and add the network definitions:

```
makedhcp -n
```

2. Define all nodes to the dhcp server:

```
makedhcp -a
```

Note: This does not add nodes that will be installed with AIX/NIM.

3. Will cause dhcp on the next request to set root-path appropriately for only node5. Note some characters (e.g. ")
must be doubly escaped (once for the shell, and once for the OMAPI layer).

```
makedhcp node5 -s 'option root-path  \"172.16.0.1:/install/freebsd6.2/x86_64\";'
```

4. Query a node from the DHCP server.

```
# makedhcp -q node01
node01: ip-address = 91.214.34.156, hardware-address = 00:00:c9:c6:6c:42
```

FILES

DHCP configuration files:

[AIX] /etc/dhcpsd.cnf

[SLES] /etc/dhcpd.conf

[RH] /etc/dhcp/dhcpd.conf

SEE ALSO

noderange(3)|noderange.3

makedns.8

NAME

makedns - sets up domain name services (DNS).

SYNOPSIS

makedns [-*h*|-***-help*]

makedns [-V|-**-verbose] [-e|--external] [*-n*|*-**-new*] [*noderange*]

makedns [-V|-**-verbose] [-e|--external] [*-d*|*-**-delete* *noderange*]

DESCRIPTION

makedns configures a DNS server on the system you run it on, which is typically the xCAT management node.

The list of nodes to include comes from either the **noderange** provided on the command line or the entries in the local /etc/hosts files.

There are several bits of information that must be included in the xCAT database before running this command.

You must set the **forwarders** attributes in the xCAT **site** definition.

The **forwarders** value should be set to the IP address of one or more nameservers at your site that can resolve names outside of your cluster. With this set up, all nodes ask the local nameserver to resolve names, and if it is a name that the MN DNS does not know about, it will try the forwarder names.

An xCAT **network** definition must be defined for each network used in the cluster. The **net** and **mask** attributes will be used by the **makedns** command.

A network **domain** and **nameservers** values must be provided either in the **network** definiton corresponding to the node or in the **site** definition.

Only entries in /etc/hosts or the hosts specified by **noderange** that have a corresponding xCAT network definition will be added to DNS.

By default, **makedns** sets up the **named** service and updates the DNS records on the local system (management node). If the -e flag is specified, it will also update the DNS records on any external DNS server that is listed in the /etc/resolv.conf on the management node. (Assuming the external DNS server can recognize the xCAT key as authentication.)

For more information on Cluster Name Resolution: Cluster_Name_Resolution

OPTIONS

-V | --verbose

> Verbose mode.

-n | --new

> Use this flag to create new named configuration and db files.

-d | --delete

> Remove the DNS records.

-e | --external

> Update DNS records to the external DNS server listed in /etc/resolv.conf.

> Enabling the site attribute *externaldns* means use 'external' DNS by default. If setting *externaldns* to 1, you need NOT use **-e** flag in every makedns call.

noderange

A set of comma delimited node names and/or group names. See the "noderange" man page for details on additional supported formats.

Examples

1

To set up DNS for all the hosts in /etc/hosts file.

makedns

2

To set up DNS for *node1*.

makedns *node1*

3

To create a new named configuration and db files for all hosts in /etc/hosts.

makedns -n

4

To delete the DNS records for *node1*.

makedns -d *node1*

SEE ALSO

makehosts(8)|makehosts.8

makehosts.8

NAME

makehosts - sets up /etc/hosts from the xCAT hosts table.

SYNOPSIS

makehosts [**-n**] [*noderange*] [**-l** | **--longnamefirst**] [**-d**] [**-m** | **--mactolinklocal**]

makehosts {**-h** | **--help**}

DESCRIPTION

makehosts updates the /etc/hosts file based on information stored in the xCAT database object definitions.

The main three bits of information needed are: node hostname, node ip and network domain name.

The hostname and ip address are specified as part of the node definition.

The domain value is taken either from the xCAT network definition associated with the node or from the cluster site definition. If you are using multiple domains in the cluster you should add the domain names to the appropriate xCAT network definition.

Note: If your node hostnames and IP addresses follow a regular pattern, you can use just a few regular expressions to generate /etc/hosts using makehosts. For details on using regular expressions see the "xcatdb" man page.

If you specify additional network interfaces in your xCAT node definitions they will also be added to the /etc/hosts file. You can specify additional network interface information (NICs) using the following node attributes: nicips, nichostnamesuffixes, nictypes, niccustomscripts, nicnetworks. You can get a description of these attributes by running "lsdef -t node -h | more" or "man nics".

OPTIONS

-n

> Completely replace the /etc/hosts file, losing any previous content. If this option is not specified, it will only replace the lines in the file that correspond to the nodes in the specified noderange.

-l | --longnamefirst

> The long name of the host will appear before the short name for each host in the /etc/hosts file. The default is short name first.

-m | --mactolinklocal

> Updates /etc/hosts file with IPv6 link local addresses, the link local address is generated from the mac address stored in mac table.

-d

> Delete rather than create records. This will also delete any additional network interfaces (NICs) included in the node definitions.

EXAMPLES

*

> Add entries to /etc/hosts for all nodes included in the xCAT node group called "compute".

```
makehosts compute
```

*

> If the xCAT hosts table contains:

```
"compute","|node(\d+)|1.2.3.($1+0)|","|(.*)|($1).cluster.net|",,
```

> Assuming the group "compute" contains node01, node02, etc., then in /etc/hosts they will be given IP addresses of 1.2.3.1, 1.2.3.2, etc.

SEE ALSO

hosts(5)|hosts.5, makedns(8)|makedns.8

makeknownhosts.8

NAME

makeknownhosts - Make a known_hosts file under $ROOTHOME/.ssh for input noderange.

SYNOPSIS

makeknownhosts *noderange* [**-r** | **--remove**] [**-V** | **--verbose**]

makeknownhosts {**-h** | **--help**}

DESCRIPTION

makeknownhosts Replaces or removes in the known_hosts file in the $ROOTHOME/.ssh directory, the enties for the nodes from the noderange input to the command. The known_hosts file entry is built from the shared ssh host key that xCAT distributes to the installed nodes.

HMCs, AMM, switches, etc., where xCAT does not distribute the shared ssh host key, should not be put in the noderange.

To build the known_hosts entry for a node, you are only required to have the node in the database, and name resolution working for the node. You do not have to be able to access the node.

Having this file with correct entries, will avoid the ssh warning when nodes are automatically added to the known_hosts file. The file should be distributed using xdcp to all the nodes, if you want node to node communication not to display the warning.

OPTIONS

noderange

>   A set of comma delimited node names and/or group names. See the "noderange" man page for details on supported formats.

-r|--remove

>   Only removes the entries for the nodes from the known_hosts file.

-V|--verbose

>   Verbose mode.

EXAMPLES

*

>   makeknownhosts compute

*

>   makeknownhosts lpars,service

*

>   makeknownhosts node02 -r

makenetworks.8

NAME

makenetworks - Gather cluster network information and add it to the xCAT database.

SYNOPSIS

makenetworks [-h|–help]

makenetworks [-v| –version]

makenetworks [-V|–verbose] [-d|–display]

DESCRIPTION

The **makenetworks** command can be used to gather network information from an xCAT cluster environment and create corresponding network definitions in the xCAT database.

Every network that will be used to install a cluster node must be defined in the xCAT database.

The default behavior is to gather network information from the managment node, and any configured xCAT service nodes, and automatically save this information in the xCAT database.

You can use the "-d" option to display the network information without writing it to the database.

You can also redirect the output to a file that can be used with the xCAT **mkdef** command to define the networks.

For example:

```
makenetworks -d > mynetstanzas

cat mynetstanzas | mkdef -z
```

This features allows you to verify and modify the network information before writing it to the database.

When the network information is gathered a default value is created for the "netname" attribute. This is done to make it possible to use the mkdef, chdef, lsdef, and rmdef commands to manage this data.

The default naming convention is to use a hyphen separated "net" and "mask" value with the "." replace by "_". (ex. "8_124_47_64-255_255_255_0")

You can also modify the xCAT "networks" database table directly using the xCAT **tabedit** command.

```
tabedit networks
```

Note: The **makenetworks** command is run automatically when xCAT is installed on a Linux management node.

OPTIONS

-d|--display Display the network definitions but do not write to the definitions to the xCAT database. The output will be in stanza file format and can be redirected to a stanza file that can be used with **mkdef** or **chdef** commands to create or modify the network definitions.

-h | --help Display usage message.

-v | --version Command Version.

-V |--verbose Verbose mode.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. Gather cluster network information and create xCAT network definitions.

```
makenetworks
```

2. Display cluster network information but do not write the network definitions to the xCAT database.

```
makenetworks -d
```

The output would be one or more stanzas of information similar to the following. The line that ends with a colon is the value of the "netname" attribute and is the name of the network object to use with the lsdef, mkdef, chdef and rmdef commands.

9_114_37_0-255_255_255_0: objtype=network gateway=9.114.37.254 mask=255.255.255.0 net=9.114.37.0

FILES

/opt/xcat/sbin/makenetworks

SEE ALSO

makedhcp(8)|makedhcp.8

makeroutes.8

NAME

makeroutes - add or delete routes to/from the os route table on nodes.

SYNOPSIS

makeroutes [**-r | --routename** *r1*[,*r2*...]]

makeroutes -d | --delete [**-r | --routenames** *r1*[,*r2*...]]

makeroutes *noderange* [**-r | --routename** *r1*[,*r2*...]]

makeroutes *noderange* **-d | --delete** [**-r | --routenames** *r1*[,*r2*...]]

makeroutes [**-h --help | -v | --version**]

DESCRIPTION

The **makeroutes** command adds or deletes routes on the management node or any given nodes. The **noderange** specifies the nodes where the routes are to be added or removed. When the *noderange* is omitted, the action will be done on the management node. The **-r** option specifies the name of routes. The details of the routes are defined in the **routes** table which contians the route name, subnet, net mask and gateway. If -r option is omitted, the names of the routes found on **noderes.routenames** for the nodes or on **site.mnroutenames** for the management node will be used.

If you want the routes be automatically setup during node deployment, first put a list of route names to **noderes.routenames** and then add *setroute* script name to the **postscripts.postbootscripts** for the nodes.

Parameters

noderange specifies the nodes where the routes are to be added or removed. If omitted, the operation will be done on the management node.

OPTIONS

-d|--delete

> Specifies to delete the given routes. If not specified, the action is to add routes.

-r|--routename

> Specifies a list of comma separated route names defined in the **routes** table. If omitted, all routes defined in **noderes.routenames** for nodes or **site.mnroutenames** for the management node will be used.

-h|--help

> Display usage message.

-v|--version

> Command Version.

EXAMPLES

1. To add all routes from the **site.mnroutenames** to the os route table for the management node.

```
makeroutes
```

2. To add all the routes from **noderes.routenames** to the os route table for node1.

```
makeroutes node1
```

3. To add route rr1 and rr2 to the os route table for the management node.

```
makeroutes -r rr1,rr2
```

4. To delete route rr1 and rr2 from the os route table on node1 and node1.

```
makeroutes node1,node2 -d -r rr1,rr2
```

FILES

/opt/xcat/sbin/makeroutes

SEE ALSO

mknb.8

NAME

mknb - creates a network boot root image for node discovery and flashing

SYNOPSIS

mknb *arch*

DESCRIPTION

The **mknb** command is run by xCAT automatically, when xCAT is installed on the management node. It creates a network boot root image (used for node discovery, BMC programming, and flashing) for the same architecture that the management node is. So you normally do not need to run the **mknb**command yourself.

If you do run **mknb** to add custom utilities to your boot root image, and you have an xCAT Hierarchical Cluster with service nodes that each have a local /tftpboot directory (site sharedtftp=0), you will also need to copy the generated root image to each service node.

Presently, only the arch x86_64 is supported.

OPTIONS

arch

> The hardware architecture for which to build the boot image: x86_64

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

SEE ALSO

makedhcp(8)|makedhcp.8

nodeadd.8

NAME

nodeadd - Adds nodes to the xCAT cluster database.

SYNOPSIS

nodeadd *noderange* **groups**=*groupnames* [*table.column=value*] [...]

nodeadd {**-v** | **--version**}

nodeadd [**-?** | **-h** | **--help**]

DESCRIPTION

The nodeadd command adds the nodes specified in noderange to the xCAT database. It also stores the any additional attributes specified for these nodes. At least one groupname must be supplied. You should also consider specifying attributes in at least the following tables: **nodehm**, **noderes**, **nodetype**. See the man page for each of these for details. Also see the **xcatdb** man page for an overview of each table.

The nodeadd command also supports some short cut names as aliases to common attributes. See the **nodels** man page for details.

OPTIONS

-v|--version

> Command Version.

-?|-h|--help

> Display usage message.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

*

> To add nodes in noderange node1-node4 with group all:
>
> **nodeadd** *node1-node4 groups=all*

*

> To add nodes in noderange node1-node4 to the nodetype table with os=rhel5:
>
> **nodeadd** *node1-node4 groups=all,rhel5 nodetype.os=rhel5*

FILES

/opt/xcat/bin/nodeadd

SEE ALSO

nodels(1)|nodels.1, nodech(1)|nodech.1, noderange(3)|noderange.3

nodeset.8

Name

nodeset - set the boot state for a noderange

Synopsis

nodeset [*noderange*] [*boot*|\ *install*\ |\ *stat*\ |\ *iscsiboot*\ |\ *netboot*\ |\ *statelite*\ |\ *offline*\ |\ *runcmd=bmcsetup*\ |\ *osimage[=<imagename*\ >]|*shell*|*shutdown*]

nodeset *noderange osimage=<imagename>* [-*-**noupdateinitrd***] [***-****-ignorekernelchk*]

nodeset [-*h*|-*-**help***|*-**v***|*-***-version*]

Description

nodeset sets the next boot state for a single or range of nodes or groups. It tells xCAT what you want to happen the next time the nodes are booted up. See noderange(3)|noderange.3. **nodeset** accomplishes this by changing the network boot files. Each xCAT node always boots from the network and downloads a boot file with instructions on what action to take next.

nodeset will manipulate the boot configuration files of yaboot and pxelinux.0.

Assume that /tftpboot is the root for tftpd (set in site(5)|site.5).

nodeset for pxe makes changes to /tftpboot/pxelinux.cfg/{node hex ip}

nodeset for yaboot makes changes to /tftpboot/etc/{node hex ip}

nodeset only sets the next boot state, but does not reboot.

nodeset is called by rinstall and winstall and is also called by the installation process remotely to set the boot state back to "boot".

A user can supply their own scripts to be run on the mn or on the service node (if a hierarchical cluster) for a node when the nodeset command is run. Such scripts are called **prescripts**. They should be copied to /install/prescripts dirctory. A table called *prescripts* is used to specify the scripts and their associated actions. The scripts to be run at the beginning of the nodeset command are stored in the 'begin' column of *prescripts* table. The scripts to be run at the end of the nodeset command are stored in the 'end' column of *prescripts* table. You can run 'tabdump prescripts -d' command for details. The following two environment variables will be passed to each script: NODES contains all the names of the nodes that need to run the script for and ACTION contains the 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 will be invoked at at a time. If it is not specified, the script will be invoked once for all the nodes.

Options

boot

> Instruct network boot loader to be skipped, generally meaning boot to hard disk

install | install=profilename | install=<os-architecture-profilename>

Prepare server for installing a node according to the table configuration. This will typically mean an unattended install file is created (i.e. kickstart or autoyast), install resources will be verified in tftp and the bulk transfer method (http, ftp, or nfs for example), and the boot loader configuration file will be written. Note: this option has been deprecated and will be removed in a future release. You should change to using the osimage option instead.

netboot | netboot=profilename | netboot=<os-architecture-profilename>

Prepare server for deploying diskless nodes. With the diskless solution, the os image will reside in the RAM of the node. Note: this option has been deprecated and will be removed in a future release. You should change to using the osimage option instead.

statelite | statelite=profilename | statelite=<os-architecture-profilename>

Prepare server for deploying statelite nodes. Statelite 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). The configuration can be done through **litefile**, **litetree** and **statelite** tables. Note: this option has been deprecated and will be removed in a future release. You should change to using the osimage option instead.

offline

Cleanup the current pxe/tftp boot configuration files for the nodes requested

osimage | osimage=<imagename>

Prepare server for installing a node using the specified os image. The os image is defined in the *osimage* table and *linuximage* table. If the <imagename> is omitted, the os image name will be obtained from *nodetype.provmethod* for the node.

--noupdateinitrd

Skip the rebuilding of initrd when the 'netdrivers', 'drvierupdatesrc' or 'osupdatename' were set for injecting new drviers to initrd. But, the geninitrd command should be run to rebuild the initrd for new drivers injecting. This is used to improve the performance of nodeset command.

--ignorekernelchk

Skip the kernel version checking when injecting drivers from osimage.driverupdatesrc. That means all drivers from osimage.driverupdatesrc will be injected to initrd for the specific target kernel.

runimage=<task>>

If you would like to run a task after deployment, you can define that task with this attribute.

stat

Display the current boot loader config file description for the nodes requested

runcmd=bmcsetup

This instructs the node to boot to the xCAT nbfs environment and proceed to configure BMC for basic remote access. This causes the IP, netmask, gateway, username, and password to be programmed according to the configuration table.

shell

This instructs tho node to boot to the xCAT genesis environment, and present a shell prompt on console. The node will also be able to be sshed into and have utilities such as wget, tftp, scp, nfs, and cifs. It will have storage drivers available for many common systems.

shutdown

To make the node to get into power off status. This status only can be used after **runcmd** and **runimage** to power off the node after the performing of operations.

-h | --help

Print help.

-v | --version

Print version.

Files

noderes table - xCAT node resources file. See noderes(5)|noderes.5 for further details.

nodetype table - xCAT node installation type file. See nodetype(5)|nodetype.5 for fur- ther details. This is used to determine the node installation image type.

site table - xCAT main configuration file. See site(5)|site.5 for further details. This is used to determine the location of the TFTP root directory and the TFTP xCAT subdirectory. /tftpboot and /tftpboot/xcat is the default.

Examples

*

To setup to install mycomputeimage on the compute node group.

nodeset compute osimage=mycomputeimage

*

To run http://\protect\T1\textdollarmaster/image.tgz after deployment:

nodeset $node runimage=http://$MASTER/image.tgznodeset

See Also

noderange(3)|noderange.3, nodels(1)|nodels.1, nodestat(1)|nodestat.1, rinstall(8)|rinstall.8, makedhcp(8)|makedhcp.8, osimage(7)|osimage.7

rescanplugins.8

NAME

rescanplugins - Notifies xcatd to rescan the plugin directory

SYNOPSIS

rescanplugins

rescanplugins {**-h | --help**}

rescanplugins {**-v | --version**}

rescanplugins [**-s | --servicenodes**]

DESCRIPTION

rescanplugins notifies the xcatd daemon to rescan the plugin directory and update its internal command handlers hash. This command should be used when plugins have been added or removed from the xCAT plugin directory (/opt/xcat/lib/perl/xCAT_plugin) or if the contents of the handled_commands subroutine in an existing plugin has changed.

If rescanplugins is called as a subrequest from another command, the xcatd command handlers hash changes will not be available to that command's process. Only subsequent command calls will see the updates.

OPTIONS

-h|--help

> Displays the usage message.

-v|--version

> Displays the release version of the code.

-s|--servicenodes

> Process the rescanplugins on the management node and on all service nodes. The rescanplugins command will be sent to the xcatd daemon on all nodes defined in the servicenode table. The default is to only run on the management node.

EXAMPLES

*

> To rescan the plugins only on the xCAT Management Node:
>
> **rescanplugins**

*

> To rescan the plugins on the xCAT Management Node and on all service nodes:
>
> **rescanplugins -s**

rinstall.8

Name

rinstall - Begin OS provision on a noderange

Synopsis

rinstall [-*o*|-*-**osver***] [*-**p***|*--**profile***] [*-**a***|*--**arch***] [*-**c***|*-******-console*] [*noderange*]

rinstall [-*O*|-*-**osimage***] [*-**c***|*-******-console*] [*noderange*]

Description

rinstall is a convenience command that will change tables as requested for operating system version, profile, and architecture, call **nodeset** to modify the network boot configuration, call **rsetboot** net to set the next boot over network (only support nodes with "nodetype.mgt=ipmi", for other nodes, make sure the correct boot order has been set before **rinstall**), and **rpower** to begin a boot cycle.

If [-O|–osimage] is specified or nodetype.provmethod=*osimage* is set, provision the noderange with the osimage specified/configured, ignore the table change options if specified.

If -c is specified, it will then run rcons on the node. This is allowed only if one node in the noderange. If need consoles on multiple nodes , see winstall(8)|winstall.8.

Options

-h | --help

> Display usage message.

-v | --version

> Display version.

-o | --osver

> Specifies which os version to provision. If unspecified, the current node os setting is used. Will be ignored if [-O|–osimage] is specified or nodetype.provmethod=*osimage*.

-p | --profile

> Specifies what profile should be used of the operating system. If not specified the current node profile setting is used. Will be ignored if [-O|–osimage] is specified or nodetype.provmethod=*osimage*.

-a | --arch

> Specifies what architecture of the OS to provision. Typically this is unneeded, but if provisioning between x86_64 and x86 frequently, this may be a useful flag. Will be ignored if [-O|–osimage] is specified or nodetype.provmethod=*osimage*.

-O | --osimage

> Specifies the osimage to provision.

-c | --console

> Requests that rinstall runs rcons once the provision starts. This will only work if there is only one node in the noderange. See winstall(8)|winstall.8 for starting nsoles on multiple nodes.

Examples

rinstall *node1-node20*

Provison nodes 1 through 20, using their current configuration.

rinstall *node1-node20* -o rhels5.1 -p compute

Provision nodes 1 through 20, forcing rhels5.1 and compute profile.

rinstall *node1-node20* -O rhels6.4-ppc64-netboot-compute

Provision nodes 1 through 20 with the osimage rhels6.4-ppc64-netboot-compute.

rinstall *node1* -c

Provisoon node1 and start a console to monitor the process.

See Also

noderange(3)|noderange.3, winstall(8)|winstall.8, rcons(1)|rcons.1

rmosdistro.8

SYNOPSIS

rmosdistro [**-a** | **--all**] [**-f**|**--force**] **osdistroname** [**osdistroname2 ...**]

rmosdistro [**-h** | **--help**]

DESCRIPTION

The **rmosdistro** command removes the specified OS Distro that was created by **copycds**. To delete all OS Distro entries, please specify **[-a|--all]**. If the specified OS Distro is referenced by some osimage, **[-f|force]** can be used to remove it.

ARGUMENTS

The OS Distro names to delete, delimited by blank space.

OPTIONS

{-a|--all}

   If specified, try to delete all the OS Distros.

{-f|--force}

   Remove referenced OS Distros, never prompt.

{-h|--help}

   Show info of rmosdistro usage.

RETURN VALUE

Zero: The command completed successfully.

Nonzero: An Error has occurred.

EXAMPLES

*

> To remove OS Distro "rhels6.2-ppc64" and "sles11.2-ppc64":
>
> **rmosdistro rhels6.2-ppc64 sles11.2-ppc64**

*

> To remove OS Distro "rhels6.2-ppc64", regardless of whether is referenced by any osimage:
>
> **rmosdistro -f rhels6.2-ppc64**

*

> To remove all OS Distros:
>
> **rmosdistro -a**

runsqlcmd.8

NAME

runsqlcmd -Runs sql command files against the current xCAT database.

SYNOPSIS

runsqlcmd

runsqlcmd {**-h | --help**}

runsqlcmd {**-v | --version**}

runsqlcmd {**-d | --dir** *directory_path*} {**-V | --verbose**}

runsqlcmd {**-f | --files** *list of files*} {**-V | --verbose**}

runsqlcmd {**-V | --verbose**} {**sql statement**}

DESCRIPTION

The runsqlcmd routine, runs the sql statements contained in the *.sql files as input to the command against the current running xCAT database. Only DB2,MySQL and PostgreSQL databases are supported. SQLite is not supported. If no directory or filelist is provided, the default /opt/xcat/lib/perl/xCAT_schema directory is used. If the directory is input with the -d flag, that directory will be used. If a comma separated list of files is input with the -f flag, those files will be used.

OPTIONS

-h|--help

> Displays the usage message.

-v|--version

> Displays current code version.

-V|--verbose

> Displays extra debug information.

-d|--dir

> To use a directory other than the default directory, enter the directory path here.

-f|--files

> Comma separated list of files (full path), wildcard (*) can be used.

File format

> The files must be of the form <name>.sql or <name>_<database>.sql where
>
> <database> is mysql,pgsql, or db2. Files must have permission 0755.

sql statement

> Quoted sql statement syntax appropriate for the current database.

EXAMPLES

*

> To run the database appropriate *.sql files in /opt/xcat/lib/perl/xCAT_schema :
>
> **runsqlcmd**

*

> To run the database appropriate *.sql files in /tmp/mysql:
>
> **runsqlcmd** *-d /tmp/mysql*

*

> To run the database appropriate *.sql files in the input list:
>
> **runsqlcmd** *-f "/tmp/mysql/test\,/tmp/mysql/test1*"***

*

> To checkout one DB2 sql file:
>
> **runsqlcmd** *-f /tmp/db2/test_db2.sql*

*

> To run the following command to the database:
>
> **runsqlcmd** *"Select \ from site;"***

setupiscsidev.8

NAME

setupiscsidev - creates a LUN for a node to boot up with, using iSCSI

SYNOPSIS

setupiscsidev [**-s|--size**] *noderange*

setupiscsidev [**-h|--help|-v|--version**]

DESCRIPTION

The **setupiscsidev** command will create a LUN on the management node (or service node) for each node specified. The LUN device can then be used by the node as an iSCSI device so the node can boot diskless, stateful.

OPTIONS

-s|--size

> The size of the LUN that should be created. Default is 4096.

-v|--version

> Display version.

-h|--help

> Display usage message.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

SEE ALSO

nodeset(8)|nodeset.8

tabch.8

NAME

tabch - Add, delete or update rows in the database tables.

SYNOPSIS

tabch [-h| –help]

tabch [-v| –version]

tabch [keycolname=keyvalue] [tablename.colname=newvalue] *

tabch [keycolname=keyvalue] [tablename.colname+=newvalue] *

tabch -d [keycolname=keyvalue] [tablename.colname=newvalue] *

DESCRIPTION

The tabch command adds, deletes or updates the attribute value in the specified table.column for the specified keyvalue. The difference between tabch and chtab is tabch runs as a plugin under the xcatd daemon. This give the additional security of being authorized by the daemon. Normally, the given value will completely replace the current attribute value. But if "+=" is used instead of "=", the specified value will be appended to the coma separated list of the attribute, if it is not already there.

OPTIONS

-h Display usage message.

-v Command Version.

-d Delete option.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. To add a node=node1 to the nodelist table with groups=all:

*tabch node=node1 nodelist.groups=all *

2. To add a keyword (tftpdir) and value (/tftpboot) to the site table:

*tabch key=tftpdir site.value=/tftpboot *

3. To add node1 to the nodetype table with os=rhel5:

tabch node=node1 nodetype.os=rhel5

4. To change node1 in nodetype table setting os=sles:

tabch node=node1 nodetype.os=sles

5. To change node1 by appending otherpkgs to the postbootscripts field in the postscripts table:

tabch node=node1 postscripts.postbootscripts+=otherpkgs

6. To delete node1 from nodetype table:

tabch -d node=node1 nodetype

FILES

/opt/xcat/sbin/tabch

SEE ALSO

tabdump(8)|tabdump.8, tabedit(8)|tabedit.8

tabdump.8

NAME

tabdump - display an xCAT database table in CSV format.

SYNOPSIS

tabdump [*-d*] [*table*]

tabdump [*table*]

tabdump [*-f filename*] [*table*]

tabdump [*-n # of records*] [*auditlog | eventlog*]

tabdump [*-w attr==val*] [**-w** *attr=~val*] ...] [*table*]

tabdump [*-w attr==val*] [**-w** *attr=~val*] ...] [*-f filename*] [*table*]

tabdump [*-v* | -********-version*]

tabdump [*-?* | *-h* | -*********-help*]

tabdump

DESCRIPTION

The tabdump command displays the header and all the rows of the specified table in CSV (comma separated values) format. Only one table can be specified. If no table is specified, the list of existing tables will be displayed.

OPTIONS

-?|-h|--help

> Display usage message.

-d

> Show descriptions of the tables, instead of the contents of the tables. If a table name is also specified, descriptions of the columns (attributes) of the table will be displayed. Otherwise, a summary of each table will be displayed.

-n

> Shows the most recent number of entries as supplied on the -n flag from the auditlog or eventlog table.

-f

> File name or path to file in which to dump the table. Without this the table is dumped to stdout. Using the -f flag allows the table to be dumped one record at a time. If tables are very large, dumping to stdout can cause problems such as running out of memory.

-w *'attr==val'* **-w** *'attr=~val'* ...

> Use one or multiple -w flags to specify the selection string that can be used to select particular rows of the table. See examples.

> 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 is greater than this ␣
→specific value.
>=        Select nodes where the attribute value is greater than or equal to␣
→this  specific value.
<         Select nodes where the attribute value is less than this  specific␣
→value.
<=        Select nodes where the attribute value is less than or equal to␣
→this  specific value.
=~        Select nodes where the attribute value matches the SQL LIKE value.
!~        Select nodes where the attribute value matches the SQL NOT LIKE␣
→value.
```

RETURN VALUE

0

The command completed successfully.

1

An error has occurred.

EXAMPLES

*

To display the contents of the site table:

tabdump site

*

To display the contents of the nodelist table where the groups attribute is compute :

tabdump -w 'groups==compute' nodelist

*

To display the contents of the nodelist table where the groups attribute is comput% where % is a wildcard and can represent any string and the status attribute is booted :

tabdump -w 'groups=~comput%' -w 'status==booted' nodelist

*

To display the records of the auditlog on date 2011-04-18 11:30:00 :

tabdump -w 'audittime==2011-04-18 11:30:00' auditlog

*

To display the records of the auditlog starting on 2011-04-18:

tabdump -w 'audittime>2011-04-18 11:30:00' auditlog

*

To display the 10 most recent entries in the auditlog:

tabdump -n 10 auditlog

*

  To see what tables exist in the xCAT database:

  tabdump

*

  To back up all the xCAT database tables, instead of running **tabdump** multiple times, you can use the **dumpxCATdb** command as follows:

  dumpxCATdb -p /tmp/xcatbak ****

  See the **dumpxCATdb** man page for details.

*

  To display a summary description of each table:

  tabdump -d

*

  To display a description of each column in the nodehm table:

  tabdump -d nodehm

FILES

/opt/xcat/sbin/tabdump

SEE ALSO

tabrestore(8)|tabrestore.8, tabedit(8)|tabedit.8, dumpxCATdb(1)|dumpxCATdb.1

tabedit.8

NAME

tabedit - view an xCAT database table in an editor and make changes.

SYNOPSIS

tabedit [*table*]

tabedit [*-?* | *-h* | -******-help*]

DESCRIPTION

The tabedit command opens the specified table in the user's editor, allows them to edit any text, and then writes changes back to the database table. The table is flattened into a CSV (comma separated values) format file before giving it to the editor. After the editor is exited, the CSV file will be translated back into the database format. You may not tabedit the auditlog or eventlog because indexes will be regenerated. Use tabprune command to edit auditlog and eventlog.

OPTIONS

-?|-h|--help

> Display usage message.

ENVIRONMENT VARIABLES

TABEDITOR

> The editor that should be used to edit the table, for example: vi, vim, emacs, oocalc, pico, gnumeric, nano. If **TABEDITOR** is not set, the value from **EDITOR** will be used. If **EDITOR** is not set, it will default to vi.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

*

> To edit the site table:

```
B<tabedit> I<site>
```

FILES

/opt/xcat/sbin/tabedit

SEE ALSO

tabrestore(8)|tabrestore.8, tabdump(8)|tabdump.8, chtab(8)|chtab.8

tabprune.8

NAME

tabprune - Deletes records from the eventlog,auditlog,isnm_perf,isnm_perf_sum tables.

SYNOPSIS

tabprune eventlog | auditlog [**-V**] **-i** *recid* |**-n** *number of records* | **-p** *percentage* | **-d** *number of days* | **-a**

tabprune tablename -a

tabprune [**-h | --help**] [**-v | --version**]

DESCRIPTION

The tabprune command is used to delete records from the auditlog,eventlog,isnm_perf,isnm_perf_sum tables. As an option, the table header and all the rows pruned from the specified table will be displayed in CSV (comma separated values) format. The all records options (-a) can be used on any xCAT table.

OPTIONS

-h|--help

> Display usage message.

-V

> Verbose mode. This will cause tabprune to display the records that are being deleted from the table, in case you want to redirect them to a file to archive them.

-a

> Remove all records from the input table name. This option can be used on any xCAT table.

-i *recid number*

> Remove the records whose recid is less than the input recid number.

-n *number*

> Remove the number of records input.

-p *percent*

> Remove the number of records input.

-d *number of days*

> Remove all records that occurred >= than number of days ago.

RETURN VALUE

0

> The command completed successfully.

1

> An error has occurred.

EXAMPLES

*

> To remove all the records in the eventlog table:
>
> **tabprune** *eventlog* -a

*

> To remove all the records in the eventlog table saving the deleted records in eventlog.csv:
>
> **tabprune** *eventlog* -V -a > eventlog.csv

*

> To remove all the records before recid=200 in the auditlog table:
>
> **tabprune** *auditlog* -i 200

*

> To remove 400 records from the auditlog table and display the remove records:
>
> **tabprune** *auditlog* -V -n 400

*

> To remove 50% of the eventlog table:
>
> **tabprune** *eventlog* -p 50

*

> To remove all records that occurred >= 5 days ago in the eventlog:
>
> **tabprune** *eventlog* -d 5

FILES

/opt/xcat/sbin/tabprune

SEE ALSO

tabrestore(8)|tabrestore.8, tabedit(8)|tabedit.8,tabdump(8)|tabdump.8

tabrestore.8

NAME

tabrestore - replaces with or adds to a xCAT database table the contents in a csv file.

SYNOPSIS

tabrestore [-a] *table.csv*

tabrestore [-*?* | -*h* | -********-help*]

tabrestore [*v* | -********-version*]

DESCRIPTION

The tabrestore command reads the contents of the specified file and puts its data in the corresponding table in the xCAT database. Any existing rows in that table are replaced unless the (-a) flag is used and then the rows in the file are added to the table. The file must be in csv format. It could be created by tabdump. Only one table can be specified.

This command can be used to copy the example table entries in /opt/xcat/share/xcat/templates/e1350 into the xCAT database.

OPTIONS

-?|-h|--help

>   Display usage message.

-v|--version

>   Display version.

-a|--addrows

>   Add rows from the CSV file to the table instead of replacing the table with the CSV file.

RETURN VALUE

0

>   The command completed successfully.

1

>   An error has occurred.

EXAMPLES

*

>   To replace the rows in the mp table with the rows in the mp.csv file:
>
>   ** tabrestore** *mp.csv*
>
>   The file mp.csv could contain something like:

```
#node,mpa,id,comments,disable
"blade","|\D+(\d+)|amm(($1-1)/14+1)|","|\D+(\d+)|(($1-1)%14+1)|",,
```

*

>   To add the rows in the mp.csv file to the rows in the mp table:
>
>   ** tabrestore** -a *mp.csv*

*

>   To restore database tables that we dumped with dumpxCATdb:
>
>   restorexCATdb -p <restore directory>

FILES

/opt/xcat/sbin/tabrestore

SEE ALSO

tabdump(8)|tabdump.8, tabedit(8)|tabedit.8, dumpxCATdb(1)|dumpxCATdb.1

winstall.8

Name

winstall - Begin OS provision on a noderange

Synopsis

winstall [-*o*|-*-**osver***] [*-**p***|*--**profile***] [*-**a***|*-**-arch*] [*noderange*]

winstall [-*O*|-***-osimage*] [*noderange*]

Description

winstall is a convenience tool that will change attributes as requested for operating system version, profile, and architecture, call **nodeset** to modify the network boot configuration, call **rsetboot** net to set the next boot over network (only support nodes with "nodetype.mgt=ipmi", for other nodes, make sure the correct boot order has been set before **winstall**), and **rpower** to begin a boot cycle.

If [-*O*|–*osimage*] is specified or nodetype.provmethod=*osimage* is set, provision the noderange with the osimage specified/configured, ignore the table change options if specified.

It will then run wcons on the nodes.

Options

-h | --help

> Display usage message.

-v | --version

> Display version.

-o | --osver

> Specifies which os version to provision. If unspecified, the current node os setting is used. Will be ignored if [-*O*|–*osimage*] is specified or nodetype.provmethod=*osimage*.

-p | --profile

> Specifies what profile should be used of the operating system. If not specified the current node profile setting is used. Will be ignored if [-*O*|–*osimage*] is specified or nodetype.provmethod=*osimage*.

-a | --arch

Specifies what architecture of the OS to provision. Typically this is unneeded, but if provisioning between x86_64 and x86 frequently, this may be a useful flag. Will be ignored if [*-O*|*–osimage*] is specified or nodetype.provmethod=*osimage*.

-O | --osimage

Specifies the osimage to provision.

Examples

winstall *node1-node20*

Provison nodes 1 through 20, using their current configuration.

winstall *node1-node20* -o rhels5.1 -p compute

Provision nodes 1 through 20, forcing rhels5.1 and compute profile.

winstall *node1-node20* -O rhels6.4-ppc64-netboot-compute

Provision nodes 1 through 20 with the osimage rhels6.4-ppc64-netboot-compute.

See Also

noderange(3)|noderange.3, rinstall(8)|rinstall.8, wcons(1)|wcons.1

xcatconfig.8

NAME

xcatconfig - Sets up the Management Node during the xCAT install.

SYNOPSIS

xcatconfig

xcatconfig {**-h | --help**}

xcatconfig {**-v | --version**}

xcatconfig {**-i | --initinstall**} [**-V | --verbose**]

xcatconfig {**-u | --updateinstall**} [**-V | --verbose**]

xcatconfig [**-k | --sshkeys**] [**-s | --sshnodehostkeys**] [**-c | --credentials**] [**-d | --database**] [**-m | --mgtnode**] [**-t | --tunables**] [**-V | --verbose**]

xcatconfig {**-f | --force**} [**-V | --verbose**]

DESCRIPTION

xcatconfig Performs basic xCAT setup operations on an xCAT management node. This command should not be run on an xCAT Service Node, unless you are making it a Management Node. See flag description below for more details.

OPTIONS

-h|--help

>   Displays the usage message.

-v|--version

>   Displays the release version of the code.

-V|--verbose

>   Displays verbose messages.

-i|--initialinstall

>   The install option is normally run as a post operation from the rpm xCAT.spec file during the initial install of xCAT on the Management Node. It will setup the root ssh keys, ssh node keys, xCAT credentials, initialize the datebase, export directories, start syslog and other daemons as needed after the initial install of xCAT.

-u|--updateinstall

>   The update install option is normally run as a post operation from the rpm xCAT.spec file during an update install of xCAT on the Management Node. It will check the setup the root ssh keys, ssh node keys, xCAT credentials, datebase, exported directories, syslog and the state of daemons needed by xCAT, after the updateinstall of xCAT. If setup is required, it will perform the operation. It will restart the necessary daemons.

-k|--sshkeys

>   This option will remove and regenerate the root id_rsa keys. It should only be used, if the keys are deleted or corrupted. The keys must then be distribute to the nodes by installing, running updatenode -k, or using xdsh -K option, for root to be able to ssh to the nodes without being prompted for a password. rspconfig will need to be run to distribute the key to the MM and HMCs. Any device, we need to ssh from the MN to the device will also have to be updated with the new ssh keys.

-s|--sshnodehostkeys

>   This option will remove and regenerate the node host ssh keys. It should only be used, if the keys are deleted or are corrupted. The keys must then be redistribute to the nodes by installing, running updatenode -k or using xdcp or pcp to copy the keys from /etc/xcat/hostkeys directory to the /etc/ssh directory on the nodes.

-c|--credentials

>   This option will remove all xcat credentials for root and any userids where credentials have been created. It will regenerate roots credentials, but the admin will have to add back all the userid credentials needed with the /opt/xcat/share/xcat/scripts/setup-local-client.sh <username> command. It should only be used, if they are deleted or become corrupted. The root credentials must be redistribed to the service nodes by installing the service node or using updatenode -k. makeconservercf must be rerun to pick up the new credentials, and conserver must be stop and started.

-d|--database

>   This option will reinitialize the basic xCAT database table setup. It will not remove any new database entries that have been added, but it is strongly suggested that you backup you database (dumpxCATdb) before using it.

-f|--force

>   The force option may be used after the install to reinitialize the Management Node. This option will re-generate keys, credential and reinititialize the site table. This option should be used, if keys or credentials

become corrupt or lost. Additional action must be taken after using the force options. ssh keys must be re-distributed to the nodes, site table attributes might need to be restored, makeconservercf needs to be rerun to pick up the new credentials and conserver stoped and started, rspconfig needs to be rerun to distribute the new keys to the MM and the HMCs. A new set of common ssh host keys will have been generated for the nodes. If you wish your nodes to be able to ssh to each other with out password intervention, then you should redistribute these new keys to the nodes. If the nodes hostkeys are updated then you will need to remove their entries from the known_hosts files on the management node before using ssh, xdsh, xdcp. Redistribute credentials and ssh keys to the service nodes and ssh keys to the nodes by using the updatenode -k command.

-m|--mgtnode

This option will add the Management Node to the database with the correct attributes set to be recognized by xCAT. This should be run after the hostname of the Management Node is set to the name that will resolve to the cluster-facing NIC.

-t|--tunables

This option will set tunable parameters on the Management and Service nodes recommended for your Linux cluster. It will only set them during initial install, if you run xcatconfig -f or xcatconfig -t.

EXAMPLES

*

To force regeneration of keys and credentials and reinitialize the site table:

xcatconfig *-f*

*

To regenerate root's ssh keys:

xcatconfig *-k*

*

To regenerate node host ssh keys:

xcatconfig *-s*

*

To regenerate node host ssh keys and credentials:

xcatconfig *-s -c*

*

To add the Management Node to the DB:

xcatconfig *-m*

xcatd.8

NAME

xcatd - The xCAT daemon

SYNOPSIS

xcatd

DESCRIPTION

The heart of the xCAT architecture is the xCAT daemon **xcatd** on the management node. This receives requests from the client, validates the requests, and then invokes the operation. The xcatd daemon also receives status and inventory info from the nodes as they are being discovered and installed/booted.

Errors and information are reported through syslog to the /var/log/messages file. You can search for xCAT in those messages.

For further information: See https://sourceforge.net/apps/mediawiki/xcat/index.php?title=XCAT_2_Architecture.

EXAMPLES

1. To start/stop/restart xcatd on Linux, enter:

```
service xcatd start

service xcatd stop

service xcatd restart
```

2. To start/stop/restart xcatd on AIX, enter:

```
restartxcatd

  or

startsrc -s xcatd

stopsrc -s xcatd
```

FILES

/opt/xcat/sbin/xcatd

SEE ALSO

xcatdebug.8

NAME

xcatdebug - Enable or disable the trace facilities for xCAT. (Only supports Linux Operating System)

xcatdebug { [**-f enable|disable** [**-c configuration file | subroutine list**]] | [**-d enable |disable**]}

DESCRIPTION

xCAT offers two trace facilities to debug the xCAT:

*

Subroutine calling trace

Display the calling trace for subroutine when it is called.

The trace message includes: The name of the called subroutine; The arguments which passed to the called subroutine; The calling stack of the subroutine. By default, the trace will be enabled to all the subroutines in the xcatd and plugin modules. The target subroutine can be configured by configuration file or through xcatdebug command line.

The flag -c is used to specify the subroutine list for subroutine calling trace, it can only work with -f. The value of -c can be configuration file: a file contains multiple lines of **SUBROUTINE_DEFINITIONsubroutine list**: **SUBROUTINE_DEFINITION | SUBROUTINE_DEFINITION|**...

SUBROUTINE_DEFINITION: is the element for the **-c** to specify the subroutine list.

The format of **SUBROUTINE_DEFINITION**: [plugin](subroutine1,subroutine2,...)

If ignoring the [plugin], the subroutines in the () should be defined in the xcatd. e.g. (daemonize,do_installm_service,do_udp_service)

Otherwise, the package name of the plugin should be specified. e.g. xCAT::Utils(isMN,Version) e.g. xCAT_plugin::DBobjectdefs(defls,process_request)

The trace log will be written to /var/log/xcat/subcallingtrace. The log file subcallingtrace will be backed up for each running of the **xcatdebug -f enable**.

*

Commented trace log

The trace log code is presented as comments in the code of xCAT. In general mode, it will be kept as comments. But in debug mode, it will be commented back as common code to display the trace log.

NOTE: This facility can be enabled by pass the **ENABLE_TRACE_CODE=1** global variable when running the xcatd. e.g. ENABLE_TRACE_CODE=1 xcatd -f

This facility offers two formats for the trace log code:

*

Trace section ## TRACE_BEGIN # print "In the debugn"; ## TRACE_END

*

Trace in a single line ## TRACE_LINE print "In the trace linen";

The **commented trace log** can be added in xcatd and plugin modules. But following section has been added into the BEGIN {} section of the target plugin module to enable the facility.

```
if (defined $ENV{ENABLE_TRACE_CODE}) {
  use xCAT::Enabletrace qw(loadtrace filter);
  loadtrace();
}
```

OPTIONS

-f

Enable or disable the **subroutine calling trace**.

For **enable**, if ignoring the **-c** flag, all the subroutines in the xcatd and plugin modules will be enabled.

For **disable**, all the subroutines which has been enabled by **-f enable** will be disabled. **-c** will be ignored.

-c

Specify the configuration file or subroutine list.

*

> **configuration file** [a file contains multiple lines of \ **SUBROUTINE_DEFINITION**]
>
> > **e.g.** (plugin_command)   xCAT_plugin::DBobjectdefs(defls,process_request) xCAT::DBobjUtils(getobjdefs)

* subroutine list** [a string like \ **SUBROUTINE_DEFINITION | SUBROUTINE_DEFINITION|...]

> **e.g.** "(plugin_command)|xCAT_plugin::DBobjectdefs(defls,process_request)|xCAT::DBobjUtils(getobjdefs)"

-d

Enable or disable the **commented trace log**.

Note: The xcatd will be restarted for the performing of **-d**.

EXAMPLES

1

> **Enable the subroutine calling trace for all the subroutines in the xcatd and plugin modules.**
> xcatdebug -f enable

2

> **Enable the subroutine calling trace for the subroutines configured in the /opt/xcat/share/xcat/samples/tracelevel0**
> xcatdebug -f enable -c /opt/xcat/share/xcat/samples/tracelevel0

3

> **Enable the subroutine calling trace for the plugin_command in xcatd and defls,process_request in the xCAT_plugin::DBo**
> xcatdebug -f enable -c "xCAT_plugin::DBobjectdefs(defls,process_request)|(plugin_command)"

4

> **Disable the subroutine calling trace for all the subroutines which have been enabled by xcatdebug -f enable.**
> xcatdebug -f disable

5

> **Enable the commented trace log xcatdebug** -d enable

6

> **Enable both the subroutine calling trace and commented trace log xcatdebug** -f enable -c
> /opt/xcat/share/xcat/samples/tracelevel0 -d enable

xcatsetup.8

NAME

xcatsetup - Prime the xCAT database using naming conventions specified in a config file.

SYNOPSIS

xcatsetup [**-s|--stanzas** *stanza-list*] [**--yesreallydeletenodes**] *cluster-config-file*

xcatsetup [**-?** | **-h** | **--help** | **-v** | **--version**]

DESCRIPTION

The **xcatsetup** command reads the specified config file that contains general information about the cluster being set up, and naming conventions and IP addresses that you want to use. It then defines the basic objects in the xCAT database representing this cluster configuration. The **xcatsetup** command prepares the database for the step of discovering the hardware that is connected to the service and cluster networks. The typical steps of setting up a system p cluster are:

*

 Install the xCAT software on the management node

*

 Create the cluster config file and run xcatsetup

*

 Put hardware control passwords in the ppchcp or ppcdirect database table

*

 Run makenetworks and makedhcp

*

 Run the discovery commands (lsslp, mkhwconn, rspconfig) as described in the System P Hardware Management cookbook.

*

 Configure and start services using makehosts, makedns, mkconserver.cf, etc.

*

 Create the images that should be installed or booted on the nodes

*

 Run nodeset and rpower/rnetboot to boot up the nodes.

The **xcatsetup** command is intended as a quick way to fill out the database for a cluster that has very regular naming patterns. The only thing it does is fill in database attributes. If your cluster does not follow consistent naming patterns, or has some other special configuration, you should define attribute values manually using mkdef(1)|mkdef.1, instead of using **xcatsetup**. The cluster config file is meant to be an easy way to prime the database; it is not meant to be a long living file that you update as the cluster changes. If you do want to run xcatsetup again at a later time, because, for example, you added a lot of nodes, you should put the total list of nodes in the config file, not just the new ones. This is because xcatsetup uses some regular expressions for groups (e.g. frame, cec, compute) that would be calculated incorrectly if the config file told xcatsetup about only the new nodes.

Speaking of regular expressions, xcatsetup creates some pretty complicated regular expressions in the database. These are useful because they keep most of the tables small, even for large clusters. But if you want to tweak them, they may be hard to understand. If after running xcatsetup, you want to convert your database to use individual rows for every node, you can do the following:

```
lsdef -z all >tmp.stanza
cat tmp.stanza | chdef -z
```

Many of the sections and attributes in the configuration file can be omitted, if you have a simple cluster, or if you want to create just 1 or 2 of the object types at this time. See the section **A Simpler Configuration File** for an example of this.

If you want to delete all of the nodes that xcatsetup created, and start over, use the **--yesreallydeletenodes** option.

Restrictions

*

> The **xcatsetup** command has only been implemented and tested for system p servers so far.

Configuration File

The **config file** is organized in stanza format and supports the keywords in the sample file below. Comment lines begin with "#". Stanzas can be ommitted if you do not want to define that type of object. The only hostname formats supported are those shown in this sample file, although you can change the base text and the numbers. For example, hmc1-hmc3 could be changed to hwmgmt01-hwmgmt12. The hostnames specified must sort correctly. I.e. use node01-node80, instead of node1-node80. This sample configuration file is for a 2 building block cluster.

```
xcat-site:
 domain = cluster.com
 # currently only direct fsp control is supported
 use-direct-fsp-control = 1
 # ISR network topology.  For example, one of the following: 128D, 64D, 32D, 16D, 8D,
 →4D, 2D, 1D
 topology = 32D
 # The nameservers in site table will be set with the value of master automatically.

xcat-service-lan:
  # IP range used for DHCP. If you set the entry, the networks table will be filled
  # automatically with this range and the dhcp interface will be set in the site
 →table.
  dhcp-dynamic-range = 50.0.0.0-50.0.0.200

xcat-hmcs:
 hostname-range = hmc1-hmc2
 starting-ip = 10.200.1.1

xcat-frames:
 # these are the connections to the frames
 hostname-range = frame[1-6]
 num-frames-per-hmc = 3
 # this lists which serial numbers go with which frame numbers
 vpd-file = vpd2bb.stanza
 # There are two rules of defining FSP/BPAs. The first defining the node's host name
 →by increasing the last bit
 # of IP address, while the second defining the node's name by varying the second bit
 →and the third bit of IP.
 # This assumes you have 2 service LANs:  a primary service LAN 10.230.0.0/255.255.0.
 →0 that all of the port 0's
 # are connected to, and a backup service LAN 10.231.0.0/255.255.0.0 that all of the
 →port 1's are connected to.
 # bpa-a-0-starting-ip = 10.230.1.1
 # bpa-b-0-starting-ip = 10.230.2.1
 # bpa-a-1-starting-ip = 10.231.1.1
```

```
 # bpa-b-1-starting-ip = 10.231.2.1
 # This assumes you have 2 service LANs:  a primary service LAN 40.x.y.z/255.0.0.0␣
 →that all of the port 0's
 # are connected to, and a backup service LAN 41.x.y.z/255.0.0.0 that all of the port␣
 →1's are connected to.
 # "x" is the frame number and "z" is the bpa/fsp id (1 for the first BPA/FSP in the␣
 →Frame/CEC, 2 for the
 # second BPA/FSP in the Frame/CEC). For BPAs "y" is always be 0 and for FSPs "y" is␣
 →the cec id.
 vlan-1 = 40
 vlan-2 = 41


xcat-cecs:
 # These are the connections to the CECs.  Either form of hostname is supported.
 #hostname-range = cec01-cec64
 hostname-range = f[1-6]c[01-12]
 # If you use the frame/cec hostname scheme above, but do not have a consistent
 # number of cecs in each frame, xcat can delete the cecs that do not get
 # supernode numbers assigned to them.
 delete-unused-cecs = 1
 # lists the HFI supernode numbers for each group of cecs in each frame
 supernode-list = supernodelist2bb.txt
 # If you do not want to specify the supernode-list at this time and you have a␣
 →consistent
 # number of cecs in each frame, you can instead just use this setting:
 num-cecs-per-frame = 12
 #fsp-a-0-starting-ip = 10.230.3.1
 #fsp-b-0-starting-ip = 10.230.4.1
 #fsp-a-1-starting-ip = 10.231.3.1
 #fsp-b-1-starting-ip = 10.231.4.1


xcat-building-blocks:
 num-frames-per-bb = 3
 num-cecs-per-bb = 32

xcat-lpars:
 num-lpars-per-cec = 8
 # If you set these, then do not set the corresponding attributes in the other node␣
 →stanzas below.
 # Except you still need to set xcat-service-nodes:starting-ip (which is the ethernet␣
 →adapter)
 #hostname-range = f[1-6]c[01-12]p[1-8]
 hostname-range = f[1-6]c[01-12]p[01,05,09,13,17,21,25,29]
 starting-ip = 10.1.1.1
 aliases = -hf0
 # ml0 is for aix.  For linux, use bond0 instead.
 otherinterfaces = -hf1:11.1.1.1,-hf2:12.1.1.1,-hf3:13.1.1.1,-ml0:14.1.1.1

xcat-service-nodes:
 num-service-nodes-per-bb = 2
 # which cecs within the bldg block that the SNs are located in
 cec-positions-in-bb = 1,32
 # this is for the ethernet NIC on each SN
 #hostname-range = sn1-sn4
 starting-ip = 10.10.1.1
 # this value is the same format as the hosts.otherinterfaces attribute except
```

```
 # the IP addresses are starting IP addresses
 #otherinterfaces = -hf0:10.10.1.1,-hf1:10.11.1.1,-hf2:10.12.1.1,-hf3:10.13.1.1,-
↪ml0:10.14.1.1

xcat-storage-nodes:
 num-storage-nodes-per-bb = 3
 # which cecs within the bldg block that the storage nodes are located in
 cec-positions-in-bb = 12,20,31
 #hostname-range = stor1-stor6
 #starting-ip = 10.20.1.1
 #aliases = -hf0
 #otherinterfaces = -hf1:10.21.1.1,-hf2:10.22.1.1,-hf3:10.23.1.1,-ml0:10.24.1.1

xcat-compute-nodes:
 #hostname-range = n001-n502
 #starting-ip = 10.30.1.1
 #aliases = -hf0
 # ml0 is for aix.  For linux, use bond0 instead.
 #otherinterfaces = -hf1:10.31.1.1,-hf2:10.32.1.1,-hf3:10.33.1.1,-ml0:10.34.1.1
```

VPD File for Frames

The **vpd-file** specifies the following vpd table attributes for the frames: node, serial, mtm, side. Use the same stanza format that accepted by the chdef(1)|chdef.1 command, as documented in xcatstanzafile(5)|xcatstanzafile.5. The purpose of this file is to enable xCAT to match up frames found through lsslp(1)|lsslp.1 discovery with the database objects created by **xcatsetup**. All of the frames in the cluster must be specified.

Here is a sample file:

```
frame1:
  objtype=node
  serial=99200G1
  mtm=9A00-100
frame2:
  objtype=node
  serial=99200D1
  mtm=9A00-100
frame3:
  objtype=node
  serial=99200G1
  mtm=9A00-100
frame4:
  objtype=node
  serial=99200D1
  mtm=9A00-100
frame5:
  objtype=node
  serial=99200G1
  mtm=9A00-100
frame6:
  objtype=node
  serial=99200D1
  mtm=9A00-100
```

Supernode Numbers for CECs

The **supernode-list** file lists what supernode numbers should be given to each CEC in each frame. Here is a sample file:

```
frame1: 0, 1, 16
frame2: 17, 32
frame3: 33, 48, 49
frame4: 64 , 65, 80
frame5: 81, 96
frame6: 97(1), 112(1), 113(1), 37(1), 55, 71
```

The name before the colon is the node name of the frame. The numbers after the colon are the supernode numbers to assign to the groups of CECs in that frame from bottom to top. Each supernode contains 4 CECs, unless it is immediately followed by "(#)", in which case the number in parenthesis indicates how many CECs are in this supernode.

A Simpler Configuration File

This is an example of a simple cluster config file that just defines the frames and CECs for 2 frames, without specifying VPD data or supernode numbers at this time.

```
xcat-site:
 use-direct-fsp-control = 1

xcat-frames:
 hostname-range = frame[1-2]

xcat-cecs:
 #hostname-range = cec[01-24]
 hostname-range = f[1-2]c[01-12]
 num-cecs-per-frame = 12


xcat-lpars:
  hostname-range = f[1-2]c[01-12]p[01,05,09,13,17,21,25,29]
```

Database Attributes Written

The following lists which database attributes are filled in as a result of each stanza. Note that depending on the values in the stanza, some attributes might not be filled in.

xcat-site

> site table: domain, nameservers, topology

xcat-hmcs

> site table: ea_primary_hmc, ea_backup_hmc
>
> nodelist table: node, groups (all HMCs (hmc)), hidden
>
> hosts table: node, ip
>
> ppc table: node, comments
>
> nodetype table: node, nodetype

xcat-frames

nodelist table: node, groups (all frames (frame)), hidden

ppc table: node, id, hcp, nodetype, sfp

nodetype table: node, nodetype

nodehm table: node, mgt

vpd table: node, serial, mtm, side

xcat-bpas

nodelist table: node, groups (bpa,all) , hidden

ppc table: node, id, hcp, nodetype, parent

nodetype table: node, nodetype

nodehm table: node, mgt

vpd table: node, serial, mtm, side

xcat-cecs

nodelist table: node, groups (all CECs (cec), all CECs in a frame (<frame>cec)), hidden

ppc table: node, supernode, hcp, id, parent

nodetype table: node, nodetype

nodehm table: node, mgt

nodegroup table: groupname, grouptype, members, wherevals (all nodes in a CEC (<cec>nodes))

nodepos: rack, u

xcat-fsps

nodelist table: node, groups (fsp,all), hidden

ppc table: node, id, hcp, nodetype, parent

nodetype table: node, nodetype

nodehm table: node, mgt

vpd table: node, serial, mtm, side

xcat-building-blocks

site table: sharedtftp, sshbetweennodes(service)

ppc table: node, parent (for frame)

xcat-service-nodes

nodelist table: node, groups (all service nodes (service), all service nodes in a BB (bb<num>service))

hosts table: node, ip, hostnames, otherinterfaces

ppc table: node, id, hcp, parent

nodetype table: node, nodetype, arch

nodehm table: node, mgt, cons

noderes table: netboot

servicenode table: node, nameserver, dhcpserver, tftpserver, nfsserver, conserver, monserver, ftpserver, nimserver, ipforward

nodegroup table: groupname, grouptype, members, wherevals (all nodes under a service node (<servicenode>nodes))

nodepos: rack, u

xcat-storage-nodes

nodelist table: node, groups (all storage nodes (storage), all storage nodes in a BB (bb<num>storage))

hosts table: node, ip, hostnames, otherinterfaces

ppc table: node, id, hcp, parent

nodetype table: node, nodetype, arch

nodehm table: node, mgt, cons

noderes table: netboot, xcatmaster, servicenode

nodepos: rack, u

xcat-compute-nodes

nodelist table: node, groups (all compute nodes (compute))

hosts table: node, ip, hostnames, otherinterfaces

ppc table: node, id, hcp, parent

nodetype table: node, nodetype, arch

nodehm table: node, mgt, cons

noderes table: netboot, xcatmaster, servicenode

nodepos: rack, u

ll-config

postscripts: postscripts

OPTIONS

-s|--stanzas *stanza-list*

A comma-separated list of stanza names that **xcatsetup** should process in the configuration file. If not specified, it will process all the stanzas that start with 'xcat' and some other stanzas that give xCAT hints about how to set up the HPC products.

This option should only be specified if you have already run **xcatsetup** earlier with the stanzas that occur before this in the configuration file. Otherwise, objects will be created that refer back to other objects that do not exist in the database.

-v|--version

Command Version.

-?|-h|--help

Display usage message.

--yesreallydeletenodes

Delete the nodes represented in the cluster config file, instead of creating them. This is useful if your first attempt with the cluster config file wasn't quite right and you want to start over. But use this option with extreme caution, because it will potentially delete a lot of nodes. If the only thing you have done so far in

your database is add nodes by running **xcatsetup**, then it is safe to use this option to start over. If you have made other changes to your database, you should first back it up using dumpxCATdb(1)|dumpxCATdb.1 before using this option.

RETURN VALUE

0 The command completed successfully.

1 An error has occurred.

EXAMPLES

1. Use the sample config.txt file at the beginning of this man page to create all the objects/nodes for a 2 building block cluster.

```
xcatsetup config.txt
```

The output:

```
Defining site attributes...
Defining HMCs...
Defining frames...
Defining CECs...
Defining building blocks...
Defining LPAR nodes...
```

2. Use the simpler config file shown earlier in this man page to create just the frame and cec objects:

```
xcatsetup config-simple.txt
```

The output:

```
Defining frames...
Defining CECs...
```

FILES

/opt/xcat/sbin/xcatsetup

SEE ALSO

mkdef(1)|mkdef.1, chdef(1)|chdef.1, lsdef(1)|lsdef.1, xcatstanzafile(5)|xcatstanzafile.5, noderange(3)|noderange.3, nodeadd(8)|nodeadd.8

xcatsnap.8

NAME

xcatsnap - Gathers information for service about the current running xCAT environment.

xcatsnap {**-h** | **--help**}

xcatsnap {-v | --version}

xcatsnap {-B | --BYPASS}

xcatsnap {-d | --dir}

DESCRIPTION

xcatsnap - The xcatsnap command gathers configuration, log and trace information about the xCAT components that are installed. This command only collects the data on the local node on which this command is run. This command is typically executed when a problem is encountered with any of these components in order to provide service information to the IBM Support Center.

This command should only be executed at the instruction of the IBM Support Center.

OPTIONS

-h|--help

> Displays the usage message.

-v|--version

> Displays the release version of the code.

-B|--bypass

> Runs in bypass mode, use if the xcatd daemon is hung.

-d|--dir

> The directory to put the snap information. Default is /tmp/xcatsnap.

ENVIRONMENT VARIABLES

EXAMPLES

*

> Run the xcatsnap routine in bypass mode and put info in /tmp/mydir :
>
> **xcatsnap** *-B -d* /tmp/mydir

*

> To run the xcatsnap routine and use default directory /tmp/xcatsnap :
>
> **xcatsnap**

Security Notices

2016 Notices

2016-11-30 - Removal of Service Stream Password

It has been brought to our attention that the xCAT product has hard-coded default passwords for the HMC/FSP to allow for IBM Service to connect to customer machines for L2/L3 support activities. This creates a security vulnerability where third parties could potentially gain root level access using these weak, hard coded passwords.

Example:

```
create_pwd => "netsDynPwdTool --create dev FipSdev",
password => "FipSdev"
```

In response, xCAT will remove these hard-coded password and interfaces from the xCAT code.

Action

No action is required for xCAT 2.12.3, and higher.

If running older versions of xCAT, update xCAT to a higher level code base that has the hard-coded default passwords removed.

The following table describes the recommended update path:

xCAT Version	Action	Release Notes
2.13, or newer	No applicable	
2.12.x	Update to **2.12.3**, or higher	2.12.3 Release Notes
2.11.x	Update to **2.12.3**, or higher	2.12.3 Release Notes
2.10.x	Update to **2.12.3**, or higher	2.12.3 Release Notes
2.9.x, or older	Update to: • **2.9.4**, or higher for **AIX** • **2.12.3**, or higher for **LINUX**	2.9.4 Release Notes