IBM PowerHA SystemMirror for AIX

Standard Edition

Version 7.2.1

PowerHA SystemMirror commands

IBM PowerHA SystemMirror for AIX

Standard Edition

Version 7.2.1

PowerHA SystemMirror commands

Note
Before using this information and the product it supports, read the information in "Notices" on page 95.

This edition applies to IBM PowerHA SystemMirror 7.2.1 Standard Edition for AIX and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this document

You can use commands to manage and configure PowerHA® SystemMirror® clusters. Each command has syntax and examples.

Highlighting

The following highlighting conventions are used in this document:

Bold	Identifies commands, subroutines, keywords, files, structures, directories, and other items whose names are predefined by the system. Bold highlighting also identifies graphical objects, such as buttons, labels, and icons that the you select.
Italics	Identifies parameters for actual names or values that you supply.
`Monospace`	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or text that you must type.

Case sensitivity in AIX

Everything in the AIX® operating system is case sensitive, which means that it distinguishes between uppercase and lowercase letters. For example, you can use the **ls** command to list files. If you type `LS`, the system responds that the command `is not found`. Likewise, **FILEA**, **FiLea**, and **filea** are three distinct file names, even if they reside in the same directory. To avoid causing undesirable actions to be performed, always ensure that you use the correct case.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

Related information

- The PowerHA SystemMirror Version 7.2.1 PDF documents are available in the PowerHA SystemMirror 7.2.1 PDFs topic.
- The PowerHA SystemMirror Version 7.2.1 release notes are available in the PowerHA SystemMirror 7.2.1 release notes topic.

PowerHA SystemMirror commands

The following commands are commonly used to obtain information about the cluster environment or to run a specific function. Each of the following commands has syntax and examples.

For complete information on a command's capabilities and restrictions, see the man page. Man pages for PowerHA SystemMirror for AIX commands are installed in the `/usr/share/man/info/EN_US/a_doc_lib/cmds/powerha_cmds` directory.

To view the man page information for a command, use the following command:

`man` *command-name*

The *command-name* is the actual name of the PowerHA SystemMirror command or script. For example, enter `man clpasswd` to obtain information about the PowerHA SystemMirror **clpasswd** command.

What's new in PowerHA SystemMirror Commands

Read about new or significantly changed information for the PowerHA SystemMirror commands topic collection.

How to see what's new or changed

In this PDF file, you might see revision bars (|) in the left margin that identifies new and changed information.

February 2018

The following information is a summary of the updates that were made to this topic collection:

- Updated information about the `add cluster` and `manage cluster` operations in the "clmgr command" on page 40 topic.

January 2018

The following information is a summary of the updates that were made to this topic collection:

- Updated information about the *Manage* attribute of an online cluster in the "clmgr command" on page 40 topic.

June 2017

The following information is a summary of the update that is made to this topic collection:

- Added the "cl_ezupdate command" on page 4 topic.

December 2016

The following information is a summary of the updates that were made to this topic collection:

- Added the `only` option in the `[ START_CAA={no|yes|only} ]` syntax that starts only CAA services. For more information, see the "clmgr command" on page 40 topic.
- Added missing flags to the following in the following topics:
  - "cl_rc.cluster command" on page 10
  - "rc.cluster command" on page 94

cl_clstop command

Purpose

Stops cluster daemons using the System Resource Controller (SRC) facility.

Syntax

```
cl_clstop [-cspoc "[-f] [-n NodeList | -g ResourceGroup]"] -f
cl_clstop [-cspoc "[-f] [-n NodeList | -g ResourceGroup]"] -g [-s] [-y] [-N | -R | -B]
cl_clstop [-cspoc "[-f] [-n NodeList | -g ResourceGroup]"] -gr [-s] [-y] [-N | -R | -B]
```

Description

The cl_clstop command shuts down cluster services across cluster nodes. By default, the **cl_clstop** command stops cluster services across all cluster nodes. However, you can specify the list of nodes on which to stop cluster services. The **cl_clstop** stops the cluster daemons by using the System Resource Controller (SRC). You stop the cluster daemons by using the gracefully option or forcefully option. The command optionally removes automatic start on reboot through the entry in the `/etc/inittab` file. You must specify a node list if the cluster daemons are being shut down by using the gracefully with takeover option. By default, the **cl_clstop** command requires that all nodes in a cluster or all nodes in a node list are accessible over the network and online; otherwise, the **cl_clstop** command fails.

Flags

-cspoc
You can use the following arguments for the C-SPOC options:

-f Forces C-SPOC command to skip default verification. If this flag is set and a cluster node is not accessible, the **cl_clstop** command reports a warning and continues execution on the other nodes.

-n NodeList
Shuts down cluster services on the nodes that are specified in the node list.

-g ResourceGroup
Generates a list of nodes that are participating in the resource group on which the **cl_clstop** command is run.

-f Forces a shutdown. Cluster daemons terminate without running any local procedures.

-g Graceful shutdown with no takeover.

-gr
Graceful shutdown with the resources that are being released by this node and taken over by another node. The daemon terminates gracefully, and the node releases its resources, which are taken over. A node list must be specified for graceful shutdown with takeover.

-s Performs a silent shutdown. This flag does not broadcast a shutdown message through **wall** command. The default setting is to broadcast.

-y Do not ask operator for confirmation before shutting down the cluster nodes. This flag is the default.

-B Stop now and on subsequent system restart.

-N Shut down now.

-R Stops on subsequent system restart and removes the entry in the `/etc/inittab` file.

Examples

1. To shut down the cluster node by using the gracefully with takeover option on node1 (releasing the resources) without sending a warning message to users before the cluster processes are stopped and resources are released, enter:

```
cl_clstop -cspoc "-n node1" -ysNgr
```

2. To forcefully and immediately shut down the cluster on all cluster nodes (resources not released) with a warning message that is broadcast to users before the cluster processes are stopped, enter:
   ```
   cl_clstop -yNf
   ```
3. To shut down the cluster node by using the gracefully on all cluster nodes options with a warning message that is broadcast to users before the cluster processes are stopped, enter:
   ```
   cl_clstop -yg
   ```

   Note: If you do not specify either the -g or -n flags, the default action occurs on all cluster nodes.

Related reference:

"clmgr command" on page 40

---

cl_convert command

Purpose

Upgrading PowerHA SystemMirror software to the newest version involves converting the Configuration Database from a previous release to that of the current release. When you install PowerHA SystemMirror, **cl_convert** is run automatically. However, if installation fails, you must run **cl_convert** from the command line. Root user privilege is required to run **cl_convert**.

Syntax

```
[-F] -v < release> [-s< simulationfile>][-i]
```

Description

The command copies the previous version's ODM data to the new version's ODM structure. If fields were deleted in the new version, the data is saved to **/tmp/cl_convert_PowerHA SystemMirror_OLD**. The command then ensures that the data is in the correct form for the new version.

When the new version is installed, the install script adds the suffix OLD to the PowerHA SystemMirrorxxx classes stored in the **/etc/objrepos** directory, and it creates the new PowerHA SystemMirrorxxx classes for the new version. The install script issues the **cl_convert** command which converts the data in PowerHA SystemMirrorxxxOLD to the corresponding new classes in PowerHA SystemMirrorxxx.

You may run the **cl_convert** command from the command line, but it is expecting the PowerHA SystemMirrorxxx and PowerHA SystemMirrorxxxOLD ODM's to already exist.

You may want to run the **cl_convert** command with the **-F** option. If the option is *not* specified, the **cl_convert** command checks for configured data in the new ODM class PowerHA SystemMirrorcluster. If data is present, the command exits without performing the conversion. If the **-F** option is specified, the command will continue without checking for present data.

Note that **cl_convert** copies the PowerHA SystemMirrorxxx and PowerHA SystemMirrorxxxOLD ODM's to a temporary file (**/tmp/tmpodmdir**) for processing before writing the final data to the PowerHA SystemMirrorxxx ODM's. If **cl_convert** encounters any kind of error, the PowerHA SystemMirrorxxx ODM's are not overwritten. If no error occurs, the PowerHA SystemMirrorxxx ODM's are overwritten and the install script will remove the PowerHA SystemMirrorxxxOLD ODM's

Note that you must be in the conversion directory to run this command:

```
/usr/es/sbin/cluster/conversion
```

Also, cl_convert assumes that the correct value for ODMDIR is set. The results of cl_convert can be found in **/tmp/clconvert.log**.

Flags

-F Force flag. Causes cl_convert to overwrite existing ODM object classes, regardless of the number of existing entries. Omitting this flag causes cl_convert to check for data in PowerHA SystemMirrorcluster (which there will always be from the previous configuration) and exit if data is encountered.

-v Release version flag. Indicates the release number of the old version.

Important: Do not use the cl_convert command unless you know the version from which you are converting.

-s <simulation_file>
Simulation flag. Indicates that instead of writing the resulting ODM data back to the new PowerHA SystemMirrorxxx ODM's, write to the specified file in text format.

-i Ignore copy flag. Specifies not to copy the PowerHA SystemMirrorxxxOLD data to the new PowerHA SystemMirrorxxx ODM's, but just operate directly on the new PowerHA SystemMirrorxxx ODM's. This is used primarily by clconvert_snapshot.

Note: The AIX environmental variable ODMDIR must be set to the directory you want to convert.

Example

If a cluster is already configured for a previous release, during the installation of a new version of PowerHA SystemMirror., the installing script will call cl_convert as:

```
cl_convert -F -v <version of prior release>
```

cl_ezupdate command

Purpose

Manages PowerHA SystemMirror and AIX software updates across the entire cluster, often without interrupting workloads that are currently running.

Syntax

```
cl_ezupdate [-v] -h
cl_ezupdate [-v] -Q {cluster|node|nim|lpp} [-N <node1,node2,...>]
cl_ezupdate [-v] {-Q {lpp|all} |-P|-A|-C|-R} [-N <node1,node2,...>]
–S <image location> [-F]
```

Description

You can use the **cl_ezupdate** command to query information about the current cluster configuration and available software updates such as AIX and PowerHA SystemMirror service packs, interim fixes, and technology levels. You can also use the **cl_ezupdate** command to preview the installation of updates and to apply or reject updates.

To use the **cl_ezupdate** tool, each node must have access to updates that you want to install. The updates can be located on a Network Installation Management (NIM) server or in a shared file system.

The **cl_ezupdate** tool provides an automatic comparison of the available updates on each of the nodes. If using NIM, all nodes must be configured to access the same lpp_source resource and contents. If the repository is a local file system directory, the local node is the reference node. The **cl_ezupdate** tool

provides an automatic copy of the local file system when the file system does not exist or is empty on any of the nodes.

Flags

-A Applies the updates that are available in the location that is specified by the **–S** flag.

-C Commits software updates to the latest installed version of PowerHA SystemMirror or the AIX operating system.

-F Forces installation of the service pack. If an interim fix has locked a fileset and if the updates are halted from installation, this flag removes the lock and installs the service pack.

Note: This flag must always be used with the **–A** flag.

-H Displays the help information for the **cl_ezupdate** command.

-Q Queries the status of the Network Installation Management (NIM) setup, cluster software, or available updates. The value option is `cluster`, `node`, `nim`, or `lpp`.

-N Specifies the node names where you want to install updates. If you specify multiple node names, you must separate each node name with a comma. By default, updates are installed on all nodes in a cluster.

-P Runs the cluster installation in `preview` mode. When you use `preview` mode, all of the installation prerequisites are checked, but updates are not installed on the system.

-R Rejects any service pack and interim fixes that are in an `applied` state.

-S Specifies location where you want the updates installed. If you specify a file system name, the path must begin with a Forward Slash key (/). If you do not specify a Forward Slash key (/), the `lpp_source` location of the NIM server will be used for installing updates.

-V Displays extended help information.

Output File

Output from the **cl_ezupdate** command is captured in the **/var/hacmp/EZUpdate/EZUpdate.log** file.

Examples

1. To display information about the NIM server, enter the following command:
   ```
   cl_ezupdate -Q nim
   ```
2. To check and display contents of updates that are available, enter the following command:
   ```
   cl_ezupdate -Q lpp -S /tmp/lppsource/inst.images
   ```
3. To install an update in an `apply` mode, enter the following command:
   ```
   cl_ezupdate -A -S HA_v720_SP1
   ```
4. To force an installation of PowerHA SystemMirror or AIX updates that are located on a NIM server and the affected filesets are locked by an interim fix, enter the following command:
   ```
   Cl_easyupdate –A –F –S HA_v720_SP1
   ```

cl_lsfs command

Purpose

Displays the characteristics of shared file systems.

Note: Arguments associated with a particular flag must be specified immediately following the flag.

Syntax

```
cl_lsfs [-cspoc"[-f] [-g ResourceGroup | -n Nodelist ]" [-q]  [-c | -l] FileSystem ]...
```

Flags

-cspoc
Argument used to specify one of the following C-SPOC options:

-f - This option has no effect when used with the **cl_lsfs** command.

-g *ResourceGroup* - Generates the list of nodes participating in the resource group where the command will be executed.

-n *nodelist* - Runs the command on this list of nodes. If more than one node, separate nodes listed by commas.

-c Specifies a different search pattern to determine if the underlying AIX **lsfs** command returned data or not.

-l Specifies that the output should be in list format.

-q Queries the logical volume manager (LVM) for the logical volume size (in 512-byte blocks) and queries the JFS superblock for the file system size, the fragment size, the compression algorithm (if any), and the number of bytes per i-node (nbpi). This information is displayed in addition to other file system characteristics reported by the **lsfs** command.

Examples

1. To display characteristics about all shared file systems in the cluster, enter:
   ```
   cl_lsfs
   ```
2. Display characteristics about the file systems shared amongst the participating nodes in *resource_grp1*.
   ```
   cl_lsfs -cspoc "-g resource_grp1"
   ```

---

cl_lsgroup command

Purpose

Displays attributes of groups that exist on a PowerHA SystemMirror cluster.

Note: Arguments associated with a particular flag must be specified immediately following the flag.

Syntax

```
cl_lsgroup [-cspoc "[-f] -g ResourceGroup | -n Nodelist"] [-c|-f] [-a | -a List ] {ALL | Group [ ,Group] ... }
```

Flags

-cspoc
Argument used to specify the following C-SPOC option:

-f - This option has no effect when used with the **cl_lsgroup** command.

-g *ResourceGroup* - Generates the list of nodes participating in the resource group where the command will be executed.

-n *nodelist* - Runs the command on this list of nodes. If more than one node, separate nodes listed by commas.

-a List
Specifies the attributes to display. The *List* parameter can include any attribute defined in the **chgroup** command, and requires a blank space between attributes. If you specify an empty list using only the -a flag, only the group names are listed.

-c Displays the attributes for each group in colon-separated records, as follows:

```
# name: attribute1: attribute2:...

Group: value1:value2:     ...
```

-f Displays the group attributes in stanzas. Each stanza is identified by a group name. Each Attribute=Value pair is listed on a separate line:

```
group:
attribute1=value
attribute2=value
attribute3=value
```

ALL | group [group]...
All resource groups, or particular group or groups to display.

Examples

1. To display the attributes of the finance group from all cluster nodes enter:

   ```
   cl_lsgroup finance
   ```

2. To display in stanza format the ID, members (users), and administrators (adms) of the finance group from all cluster nodes, enter:

   ```
   cl_lsgroup -f -a id users adms finance
   ```

3. To display the attributes of all the groups from all the cluster nodes in colon-separated format, enter:

   ```
   cl_lsgroup -c ALL
   ```

---

cl_lslv command

Purpose

Displays shared logical volume attributes.

Note: Arguments associated with a particular flag must be specified immediately following the flag.

Syntax

```
cl_lslv [-cspoc "[-f] [-g ResourceGroup | -n Nodelist "] ]  [-l | -m] LogicalVolume
```

Flags

-cspoc
Argument used to specify one of the following C-SPOC options:

-f - This option has no effect when used with the **cl_lsfs** command.

-g *ResourceGroup* - Generates the list of nodes participating in the resource group where the command will be executed.

-n *Nodelist* - Runs the command on this list of nodes. If more than one node, separate nodes listed by commas.

-l *Logical Volume*
Lists information for each physical volume in the shared logical volume. Refer to the **lslv** command for information about the fields displayed.

-m *Logical Volume*
Lists information for each logical partition. Refer to the **lslv** command for information about the fields displayed. If no flags are specified, information about the shared logical volume and its underlying shared volume group is displayed. Refer to the **lslv** command for the information about the fields displayed.

Examples

1. To display information about the shared logical volume *lv03* , enter:

   ```
   cl_lslv -cspoc -g resource_grp1 lv03
   ```

   Information about logical volume *lv03* , its logical and physical partitions, and the volume group to which it belongs is displayed.
2. To display information about a specific logical volume, using the identifier, enter:

   ```
   cl_lslv -g resource_grp1 00000256a81634bc.2
   ```

   All available characteristics and status of this logical volume are displayed.

---

cl_lsuser command

Purpose

Displays user account attributes for users that exist on a PowerHA SystemMirror cluster.

Note: Arguments associated with a particular flag must be specified immediately following the flag.

Syntax

```
cl_lsuser [-cspoc "[-f] [-g ResourceGroup | -n Nodelist]"]  [-c | -f] [-a List ] {ALL | Name [ ,Name ]... }
```

Flags

-cspoc
Argument used to specify the following C-SPOC option:

-f - This option has no effect when used with the **cl_lsuser** command.

-g *ResourceGroup* - Generates the list of nodes participating in the resource group where the command will be executed.

-n *Nodelist* - Runs the command on this list of nodes. If more than one node, separate nodes listed by commas.

-a *Lists*
Specifies the attributes to display. The List variable can include any attribute defined in the **chuser** command and requires a blank space between attributes. If you specify an empty list, only the user names are displayed.

-c Displays the user attributes in colon-separated records, as follows:

```
# name:  attribute1:  attribute2:  ...
User:      value1:      value2:      ...
```

-f Displays the output in stanzas, with each stanza identified by a user name. Each Attribute=Value pair is listed on a separate line:

```
user:
attribute1=value
attribute2=value
attribute3=value
```

ALL | Name [name]...
Display information for all users or specified user or users.

Examples

1. To display in stanza format the user ID and group-related information about the *smith* account from all cluster nodes, enter:

   ```
   cl_lsuser -fa id pgrp groups admgroups smith
   ```
2. To display all attributes of user *smith* in the default format from all cluster nodes, enter:

   ```
   cl_lsuser smith
   ```

3. To display all attributes of all the users on the cluster, enter:
   ```
   cl_lsuser ALL
   ```

---

cl_lsvg command

Purpose

Displays information about shared volume groups.

Note: Arguments associated with a particular flag must be specified immediately following the flag.

Syntax

```
cl_lsvg [-cspoc "[-f] [-g ResourceGroup | n- Nodelist ]" [-o] |[-l | -M | -p] Volume Group...INFO HERE
```

Flags

-cspoc
Argument used to specify one of the

-f - This option has no effect when used with the **cl_lsvg** command.

-g *ResourceGroup* - Specifies the name of the resource group whose participating nodes share the volume group. The command executes on these nodes.

-n *Nodelist* - Runs the command on this list of nodes. If more than one node, separate nodes listed by commas.

-p Lists the following information for each physical volume within the group specified by the **VolumeGroup** parameter:

- Physical volume: A physical volume within the group.

- PVstate : State of the physical volume.

- **Total PPs** : Total number of physical partitions on the physical volume.

- Free PPs : Number of free physical partitions on the physical volume.

- **Distribution** : The number of physical partitions allocated within each section of the physical volume: outer edge, outer middle, center, inner middle, and inner edge of the physical volume.

-l Lists the following information for each logical volume within the group specified by the *VolumeGroup* parameter:

- LV : A logical volume within the volume group.

- Type : Logical volume type.

- LPs : Number of logical partitions in the logical volume.

- PPs: Number of physical partitions used by the logical volume.

- PVs : Number of physical volumes used by the logical volume.

-M Lists the following fields for each logical volume on the physical volume:

- PVname: *PPnum* **[** *LVname* **:** *LPnum* **[:** *Copynum* **] [** *PPstate* **]]**

- **PVname** : Name of the physical volume as specified by the system.

- **PPnum** : Physical partition number. Physical partition numbers can range from 1 to 1016.

-o Lists only the active volume groups (those that are varied on). An active volume group is one that is available for use. Refer to the **lsvg** command for the information displayed if no flags are specified.

Examples

1. To display the names of all shared volume groups in the cluster, enter:

   ```
   cl_lsvg
   nodeA: testvg
   nodeB: testvg
   ```

2. To display the names of all active shared volume groups in the cluster, enter:

   ```
   cl_lsvg -o
   nodeA: testvg
   ```

3. To display information about the shared volume group *vg02* , enter:

   ```
   cl_lsvg -cspoc testvg
   ```

---

cl_nodecmd command

Purpose

Runs a given command in parallel on a given set of nodes.

Syntax

```
cl_nodecmd [-q] [-cspoc "[-f] [-n nodelist | -g resourcegroup ]" ] command args
```

Flags

-q Specifies quiet mode. All standard output is suppressed.

-cspoc
Argument used to specify one of the following C-SPOC options:

-f - Forces **cl_nodecmd** to skip PowerHA SystemMirror version compatibility checks and node accessibility verification.

-g *resource group* - Generates the list of nodes participating in the resource group where the command will be executed.

-n *nodelist* - Runs the command on this list of nodes. If more than one node, separate nodes listed by commas.

command
Specifies the command to be run on all nodes in the nodelist.

args
Specifies arguments that are passed to the **cl_nodecmd** command.

Examples

1. Run the **lspv** command on all cluster nodes.

   ```
   cl_nodecmd lspv
   ```

2. Runs the **lsvg rootvg** command on nodes *beaver* and *dam* , suppressing standard output.

   ```
   cl_nodecmd -cspoc "-n beaver,dam" lsvg rootvg
   ```

---

cl_rc.cluster command

Purpose

Sets up the operating system environment and starts the cluster daemons across cluster nodes.

Syntax

```
cl_rc.cluster [-cspoc "[-f] [-g ResourceGroup | -nNodeList ]"] [-boot]
[b] [-i | I] [-N | -R | -B] [-M | -A] [-x] [-r] [-v] [-C interactive|yes]
```

Note: Arguments associated with a particular flag must be specified immediately following the flag.

Flags

-cspoc
Argument used to specify the following C-SPOC option:

-f - Forces **cl_rc.cluster** to skip PowerHA SystemMirror version compatibility checks and node accessibility verification.

-g *ResourceGroup* - Specifies the name of the resource group whose participating nodes share the volume group. The command executes on these nodes.

-n *Nodelist* - Executes underlying AIX commands across nodes in the nodelist.

-boot
Configures the service network interface to be on its boot address if IPAT is enabled.

-i Starts the Cluster Information (**clinfoES**) daemon with its default options.

-I Starts the Cluster Information (**clinfoES**) daemon with traps enabled.

-b Broadcasts the startup.

-N Starts the daemons immediately (no **inittab** change).

-R Starts the PowerHA SystemMirror daemons on system restart only (the PowerHA SystemMirror startup command is added to **inittab** file).

-B Starts the daemons immediately and adds the PowerHA SystemMirror entry to the **inittab** file.

-C Specifies the mode to use for corrective action when a problem occurs. Specify **yes** to automatically correct problems. Specify **interactive** to be prompted before each corrective action is run.

-M Starts the cluster services with Manual resource acquisition mode. Use this option if you want to bring the resource groups online manually.

-A Starts the cluster services with Automatic resource acquisition mode. Use this option if you want to bring resource groups online automatically on cluster startup. This is the default option.

-f Forced startup. Cluster daemons should initialize running local procedures.

-r Reacquires cluster resources after a forced down. Use this option if you changed the state of any cluster resources (ip labels, disks, applications) while the cluster was forced down.

-v Ignore verification errors during startup (auto ver sync)

-x Activates NFS cross-mounts.

Examples

1. To start the cluster with **clinfo** running on all the cluster nodes, run the following command:
   ```
   cl_rc.cluster -boot -i
   ```
2. To start the cluster with **clinfo** running on all the cluster nodes with traps enabled, run the following command:
   ```
   cl_rc.cluster -boot -I
   ```

---

clconvert_snapshot command

Purpose

The command copies the previous version's ODM data from the snapshot_file to the format of the new version's ODM structure.

Syntax

```
clconvert_snapshot -v release -s < snapshotfile >
```

Description

You can run **clconvert_snapshot** to upgrade cluster snapshots from a previous version of PowerHA SystemMirror to the most recent version of PowerHA SystemMirror. The command by default assumes you are converting to the latest version of the software.

If fields were deleted in the new version, the data is saved to */tmp/cl_convert_PowerHA SystemMirror_OLD* . The command then ensures that the data is in the correct form for the new version.

Once a snapshot file has been upgraded, it is assigned the same name as the previous version and cannot be reverted back to the previous version. A copy of the old version of the snapshot will be saved for you with the same original name plus the *.old* extension.

You must be in the **/usr/es/sbin/cluster/conversion** directory on the same node that took the snapshot to run the `clconvert_snapshot` command.

Once the snapshot file has been upgraded and all of the nodes in the cluster have the current level installed, the upgraded snapshot can be applied and then the cluster can be brought up.

The script `clconvert_snapshot` creates an old version of the ODMs and populates those ODMs with the values from the user-supplied snapshot file. It then calls the same commands that *cl_convert* uses to convert those ODMs to the current version. A new snapshot is taken from the upgraded ODMs and copied to the user supplied snapshot file.

The **clconvert_snapshot** is *not* run automatically during installation, and must always be run from the command line.

Table 1. clconvert_snapshot flags

Flag	Description
-v	Release version flag. Specifies the release number from which the conversion is to be performed. **Important:** Do not use the clconvert_snapshot command unless you know the version from which you are converting.
-s	Snapshot file flag. Specifies the snapshot file to convert. If you do not specify a path, for the snapshot file, the command uses the path specified in the **$SNAPSHOTPATH** variable. The default is **/usr/es/sbin/cluster/snapshots**.

Example

Run the following command to convert a PowerHA SystemMirror 5.3 snapshot to a current PowerHA SystemMirror snapshot named "mysnapshot."

```
clconvert_snapshot -v 5.3 -s mysnapshot
```

The file "mysnapshot" is in turn placed in the directory specified by the **$SNAPSHOTPATH** environment variable. If a **$SNAPSHOTPATH** variable is *not* specified, the file is put in **/usr/es/sbin/cluster/snapshots**.

clcheck_server command

Purpose

Returns status of daemons in a PowerHA SystemMirror cluster.

Syntax

```
clcheck_server daemon
```

Description

The **clcheck_server** command returns the status of the named daemon. This command is for use within shell scripts that need to reliably determine the status of a daemon. This command makes extra checks beyond what is done by the **lssrc** command that is provided by the System Resource Controller (SRC).

Before you use the **clcheck_server** command, you must understand the purpose of the daemon that is being checked.

Flags

daemon
Specifies the name of the daemon that you want to check.

Example

To check the status of the `clinfo` daemon, enter:

```
    if ! clcheck_server clinfoES
    then
 echo "clinfo is active"
    else
 echo "clinfo is inactive"
    fi
```

clfindres command

Purpose

Finds a given resource group or groups in a cluster configuration.

Syntax

```
clfindres [-s] [resgroup1] [resgroup2]...
```

Description

When you run **clfindres** , it calls **clRGinfo,** and the command output for **clfindres** is the same as it is for the **clRGinfo** command. Therefore, use the **clRGinfo** command to find the status and the location of the resource groups. The **-s** flag for the **clfindres** command requests abbreviated (location only) output. See the **clRGinfo** command for more information.

clgetactivenodes command

Purpose

Retrieves the names of all cluster nodes.

Syntax

```
clgetactivenodes [-n nodename ] [-o odmdir ] [-ttimeout ]  [-v verbose ]
```

Table 2. clgetactivenodes flags

Flag	Description
-n nodename	Determines if the specified node is active.
-o odmdir	Specifies *odmdir* as the ODM object repository directory instead of the default **/etc/objrepos**.
-t timeout	Specifies a maximum time interval for receiving information about active nodes.
-v verbose	Specifies that information about active nodes be displayed as verbose output.

Example

Run the following command to verify that node *java* is active.

```
clgetactivenodes -n java
```

clgetaddr command

Purpose

Returns an address that you can ping for the specified node name.

Syntax

```
clgetaddr [-o odmdir ] nodename
```

-o Specifies an alternate ODM directory.

Example

To get a PINGable address for the node *seaweed*, enter:

```
clgetaddr seaweed
```

The following address is returned: `2361059035`

cli_assign_pvids command

Purpose

Assigns a PVID to each of the disks that are passed as arguments, then update all other cluster nodes with those PVIDs.

Syntax

```
cli_assign_pvids  PhysicalVolume ...
```

Description

The Logical Volume Manager (LVM) assigns a PVID to each of the physical volumes in the list (if one is not already present), and then makes those PVIDs known on all cluster nodes.

Example

To assign PVIDs to a list of disks and have those PVIDs known across the cluster, enter:

```
cli_assign_pvids hdisk101 hdisk102 hdisk103
```

cli_chfs command

Purpose

Change the attributes of a file system on all nodes in a cluster.

Syntax

```
cli_chfs [ -m NewMountPoint ] [ -u MountGroup ] [ -p { ro | rw  } ]
         [ -t { yes | no } ] [ -a Attribute=Value ] [ -d Attribute ]
   FileSystem
```

Description

Uses C-SPOC to run the **chfs** command with the parameters, and update file system definition on all cluster nodes.

Flags

-d Attribute
Deletes the specified attribute from the `/etc/filesystems` file for the specified file system.

-m NewMountPoint
Specifies a new mount point for the specified file system. The following values are valid:

-p Sets the permissions for the file system. The following values are valid:

ro Specifies read-only permissions.

rw Specifies read-write permissions.

-t Sets the accounting attribute for the specified file system. The following values are valid:

yes
File system accounting is processed by the accounting subsystem.

no File system accounting is not processed by the accounting subsystem. This is the default value.

-u MountGroup
Specifies the mount group. Mount groups are used to group related mounts so that they can be mounted as one group instead of mounting each individually. For example, when you perform certain tests, if several scratch file systems are required to be mounted together, they can each be placed in the test mount group. You can mount this mount group with a single command, such as the **mount -t** command.

-a Attribute=Value
Specifies the `Attribute=Value` pairs dependent on virtual file system type. To specify more than one Attribute=Value pair, provide multiple -a Attribute=Value parameters.

Example

To change the size of the shared file system that is named */test_fs*, enter:

```
cli_chfs -a size=32768 /test_fs
```

Related information:

chfs command

cli_chlv command

Purpose

Change the attributes of a logical volume on all nodes in a cluster.

Syntax

```
cli_chlv  [-a Position] [-b BadBlocks] [-d Schedule] [-e Range]
          [-L label] [-p Permission] [-r Relocate] [-s Strict]
          [-t Type] [-u Upperbound] [-v Verify] [-w MirrorWriteConsistency]
          [-x Maximum] [-U userid] [-G groupid] [-P modes] LogicalVolume
```

Description

Uses C-SPOC to run the **chlv** command with specified the parameters, and update the logical volume definition on all cluster nodes.

Flags

-a Position
Sets the physical volume allocation policy (the position of the logical partitions on the physical volume). The following *Position* variables are valid:

- **m** Allocates logical partitions in the outer middle section of each physical volume. This variable is the default setting.
- **c** Allocates logical partitions in the center section of each physical volume.
- **e** Allocates logical partitions in the outer edge section of each physical volume.
- **ie** Allocates logical partitions in the inner edge section of each physical volume.
- **im** Allocates logical partitions in the inner middle section of each physical volume.

-b BadBlocks
Sets the bad-block relocation policy. The following *BadBlocks* variables are valid:

- **y** Causes bad-block relocation to occur.
- **n** Prevents bad block relocation from occurring.

-d Schedule
Sets the scheduling policy when more than one logical partition is written. You must use parallel or sequential processing to mirror a striped logical volume. The following *Schedule* variables are valid:

- **p** Establishes a parallel scheduling policy.
- **ps** Parallel write with sequential read policy. All mirrors are written in parallel but always read from the first mirror if it is available.
- **pr** Parallel write and reads are done for all mirrors. This policy is similar to the parallel policy, except an attempt is made to spread the reads to the logical volume more evenly across all mirrors.
- **s** Establishes a sequential scheduling policy. Use this variable when you specify policy of parallel or sequential strictness (super strictness).

-e Range
Sets the physical volume allocation policy. The allocation policy is the number of physical volumes to extend across by using the volumes that provide the best allocation. The value of the *Range* variable is limited by the *Upperbound* variable that is set with the **-u** flag. The following *Range* variables are valid:

- **x** Allocates logical partitions across the maximum number of physical volumes.

m Allocates logical partitions across the minimum number of physical volumes.

-G Groupid
Specifies group ID for the logical volume special file.

-L Label
Sets the logical volume label. The maximum size for this variable is 127 characters.

-n NewLogicalVolume
Changes the name of the logical volume that is specified by the *NewLogicalVolume* variable. Logical volume names must be unique system wide and can have a maximum of 15 characters.

-p Permission
Sets the access permission to read/write or read-only. The following *Permission* variables are valid:

w Sets the access permission to read/write.

r Sets the access permission to read-only. Mounting a JFS file system on a read-only logical volume is not supported.

-P Modes
Specifies permissions (file modes) for the logical volume special file.

-r Relocate
Specifies whether you want to allow or prevent the relocation of the logical volume during reorganization. The following *Relocate* variables are valid:

y Allows the logical volume to be relocated during reorganization. If the logical volume is striped, you cannot use the **chlv** command to change the relocation flag to `y`.

n Prevents the logical volume from being relocated during reorganization.

-s Strict
Determines the strict allocation policy. You can allocate copies of a logical partition to be shared or not shared for the same physical volume. The following *Strict* variables are valid:

y Sets a strict allocation policy, so copies of a logical partition cannot share the same physical volume.

n Does not set a strict allocation policy, so copies of a logical partition can share the same physical volume.

s Sets a super strict allocation policy, so that the partitions allocated for one mirror cannot share a physical volume with the partitions from another mirror. When you change to a non-super strict logical volume to a super strict logical volume, you must use the `-u` flag.

-t Type
Sets the logical volume type. The maximum size is 31 characters. If the logical volume is striped, you cannot change the *Type* variable to boot.

-U Userid
Specifies user ID for the logical volume special file.

-u Upperbound
Sets the maximum number of physical volumes for the new allocation. The value of the *Upperbound* variable is between one and the total number of physical volumes. When you use super strictness, the upper bound indicates the maximum number of physical volumes that are allowed for each mirror copy. When you use striped logical volumes, the upper bound must be multiple of *Stripe_width* variable.

-v Verify
Sets the write-verify state for the logical volume. Causes all writes to the logical volume either to be verified with a follow-up read or not to be verified with a follow-up read. The following *Verify* variables are valid:

y All writes to the logical volume are verified with a follow-up read.

n All writes to the logical volume are not verified with a follow-up read.

-w MirrorWriteConsistency
The following *MirrorWriteConsistency* variables are valid:

y Turns on active mirror write consistency. This variable verifies data consistency on the mirrored copies of a logical volume during normal I/O processing.

p Turns on passive mirror write consistency. This variable verifies data consistency on the mirrored copies during volume group synchronization after a system interruption. This function is only available on Big Volume Groups.

n No mirror write consistency.

-x Maximum
Sets the maximum number of logical partitions that can be allocated to the logical volume. The maximum number of logical partitions per logical volume is 32,512.

Example

To change the physical volume allocation of logical volume that is named *lv01*, enter:

```
cli_chlv -e m lv01
```

Related information:

chlv command

---

cli_chvg command

Purpose

Change the attributes of a volume group on all nodes in a cluster.

Syntax

```
cli_chvg  [ -s Sync { y | n }] [ -L LTGSize ] [ -Q { n | y } ] [ -u ]
          [ -t [factor ] ] [ -M { y | n | s } ] [ -B ] [ -C  ] VolumeGroup
```

Description

Uses C-SPOC to run the **chvg** command with the specified parameters and make the updated volume group definition available on all cluster nodes.

Flags

-B Changes the volume group to Big VG format. This flag can accommodate up to 128 physical volumes and 512 logical volumes. You cannot use this flag if there are any stale physical partitions in the cluster node. To use this flag, you must have enough free partitions available on each physical volume for the VGDA expansion.

Because the VGDA resides on the edge of the disk and it requires contiguous space for expansion, the free partitions are required on the edge of the disk. If those partitions are allocated for application data, they are migrated to other free partitions on the same disk. The rest of the physical partitions are renumbered to reflect the loss of the partitions for VGDA usage. This process changes the mappings of the logical to physical partitions in all the physical volumes in the volume group.

If you saved the mappings of the logical volumes for a potential recovery operation, you can generate the maps again after the completion of the conversion operation. If the backup of the volume group is taken with the map option and if you plan to restore using those maps, the restore operation can

fail since the partition number might not exist (due to reduction). It is recommended that you back up your logical volumes before you start the conversion process, and right after the conversion process completes if you use the map option.

Because the VGDA space is increased substantially, every VGDA update operation (creating a logical volume, changing a logical volume, adding a physical volume, and so on) might take considerably more time to run.

-C Changes the volume group into an enhanced concurrent capable volume group. Changes the volume group from a non-concurrent mode (varied on) to enhanced concurrent mode. This process requires that the volume group is reimported on all other nodes before activation of the enhanced concurrent mode. Changes the volume group from a concurrent mode (varied on) to an enhanced concurrent mode. You can use this flag only in a PowerHA SystemMirror cluster, and the cluster must be configured before you activate an enhanced concurrent volume group.

-L LTGSize
Sets the logical track group size to the common max transfer size of the disks when a volume group is varied on. The value of the *LTGSize* variable must be 0, 128, 256, 512, or 1024. The *LTGSize* variable must be less than or equal to the maximum transfer size of all disks in the volume group. The default value for the *LTGSize* variable is 128. If you specify an *LTGSize* of 0, the **varyonvg** command sets the logical track group size to the common max transfer size of the disks.

-M Changes the mirror pool structures for the volume group. The following values are valid:

y Each logical volume copy that is created in the volume group must be assigned to a mirror pool.

n Restrictions are not placed on the user of the mirror pool. This option is the default value.

s Super-strict mirror pools are enforced on the volume group.

Note: Local and remote physical volumes cannot belong to the same mirror pool. A volume group can contain a maximum of three mirror pools. Each mirror pool must contain at least one copy of each logical volume in the volume group.

-Q Determines whether the volume group is automatically varied off after losing its quorum of physical volumes. The default value is yes. The change takes effect the next time the volume group is activated. The following values are valid:

y The volume group is automatically varied off after losing its quorum of physical volumes.

n The volume group is active until it loses all of its physical volumes.

-s Sync
Sets the synchronization characteristics for the volume group that is specified by the *VolumeGroup* variable. This flag does not affect non-mirrored logical volumes. This flag is not supported for concurrent capable volume groups.

Automatic synchronization is a recovery mechanism that is only attempted after the Logical Volume Manager (LVM) device driver logs LVM_SA_STALEPP in the AIX operating system error log. A partition that becomes stale through any other path (for example, the **mklvcopy** command) is not automatically resynced. The following values are valid:

y Attempts to automatically synchronize stale partitions.

n Prohibits automatic synchronization of stale partitions. This value is the default setting for a volume group.

-t factor
Changes the limit of the number of physical partitions per physical volume, which is specified by a factor. The factor must be 1 - 16 for 32 physical volumes groups. The factor must be 1 - 64 for 128 physical volume groups.

If you do not specify the factor, it is set to the lowest value such that the number of physical partitions of the largest disk in the volume group is less than the factor value multiplied by 1016.

If you do specify a factor, the maximum number of physical partitions per physical volume for the volume group changes to the factor value multiplied by 1016.

Review the following information when you are determining the value for the factor:

- This flag is ignored for scalable-type volume groups.
- This flag cannot be used if the volume group is varied on in concurrent mode.
- The factor value cannot be changed if there are any stale physical partitions in the volume group.
- The maximum number of physical volumes that are allowed in this volume group is reduced to the value you get when you divide MAXPVS by the factor value (MAXPVS/factor).
- Changing an existing volume group to scalable volume group format, modifies the device subtype (reported by the `IOCINFO ioctl()` call) for all associated logical volumes to *DS_LVZ*, regardless of the previous subtype. This change does not alter any behavior of the logical volumes beyond the reported subtype.

-u Unlocks the volume group. This flag is available if the volume group is left in a locked state by an abnormal termination of another LVM operation (such as the command core dumping or the system crashing). Before you can use this flag, you must verify that the volume group is not being used by another LVM command.

Example

To turn off quorum for a volume group that is named *vg01*, enter:

```
cli_chvg -Q n vg01
```

Related information:

chvg command

---

cli_crfs command

Purpose

Create a new file system and make it available on all nodes in a cluster

Syntax

```
cli_crfs  -v VfsType { -g VolumeGroup | -d Device } [ -l LogPartitions ]
    -m MountPoint [ -u MountGroup ] [ -A { yes | no } ]
    [ -p {ro | rw } ] [ -a Attribute=Value ... ] [ -t { yes | no } ]
```

Description

Uses C-SPOC to run the **crfs** command with the specified parameters, and make the updated file system definition available on all cluster nodes.

Flags

-a Attribute=Value
Specifies a virtual file system-dependent attribute and value pair. To specify more than one attribute and value pair, provide multiple -a Attribute=Value parameters.

-d Device
Specifies the device name of a device or logical volume on which to make the file system. This flag is used to create a file system on a logical volume that exists.

-g VolumeGroup
Specifies an existing volume group on which to make the file system. A volume group is a collection of one or more physical volumes.

-l LogPartitions
Specifies the size of the log logical volume, expressed as a number of logical partitions. This flag applies only to JFS and JFS2 file systems that do not have a log device.

-m MountPoint
Specifies the mount point, which is the directory where the file system is made available. If you specify a relative path name, it is converted to an absolute path name before it is inserted into the `/etc/filesystems` file.

-p Sets the permissions for the file system.

ro Read-only permissions

rw Read/write permissions

-t Specifies whether the file system is processed by the accounting subsystem. The following values are valid:

yes
Accounting is enabled on the file system.

no Accounting is not enabled on the file system. This value is the default setting.

-u MountGroup
Specifies the mount group.

-v VfsType
Specifies the virtual file system type. The *agblksize* attribute is set when you create the file system and cannot be changed after the file system is created. The *size* attribute defines the minimum file system size. You cannot decrease the file system size with the *size* attribute after the file system is created.

Example

To create a JFS file system on an existing logical volume that is named *lv01*, enter:

```
cli_crfs -v jfs -d lv01 -m /tstvg -a 'size=32768'
```

Related information:

crfs command

---

cli_crlvfs command

Purpose

Create a new logical volume and file system, and make it available on all nodes in a cluster

Syntax

```
cli_crlvfs -v VfsType -g VolumeGroup  [ -l LogPartitions ] -m MountPoint
        [ -u MountGroup ] [ -A { yes | no } ] [ -p {ro | rw } ]
  [ -a Attribute=Value ... ] [ -t { yes | no } ]
```

Description

Uses C-SPOC to run the **crfs** command with the specified parameters, and make the updated file system definition available on all cluster nodes.

Flags

-a Attribute=Value
Specifies a virtual file system-dependent attribute and value pair. To specify more than one attribute and value pair, provide multiple -a Attribute=Value parameters.

-g VolumeGroup
Specifies an existing volume group on which to make the file system. A volume group is a collection of one or more physical volumes.

-l LogPartitions
Specifies the size of the log logical volume, expressed as a number of logical partitions. This flag applies only to JFS and JFS2 file systems that do not have a log device.

-m MountPoint
Specifies the mount point, which is the directory where the file system is made available. If you specify a relative path name, it is converted to an absolute path name before it is inserted into the `/etc/filesystems` file.

-p Sets the permissions for the file system.

ro Read-only permissions

rw Read/write permissions

-t Specifies whether the file system is processed by the accounting subsystem. The following values are valid:

yes
Accounting is enabled on the file system.

no Accounting is not enabled on the file system. This value is the default setting.

-u MountGroup
Specifies the mount group.

-v VfsType
Specifies the virtual file system type. The *agblksize* attribute is set when you create the file system and cannot be changed after the file system is created. The *size* attribute defines the minimum file system size. You cannot decrease the file system size with the *size* attribute after the file system is created.

Example

To create a JFS file system on a volume group that is named *vg01*, enter:

```
cli_crlvfs -v jfs -g vg01 -m /tstvg -a 'size=32768'
```

---

cli_extendlv command

Purpose

Increases the size of a logical volume on all nodes in a cluster by adding unallocated physical partitions from within the volume group.

Syntax

```
cli_extendlv [ -a Position ] [ -e Range ] [ -u Upperbound ] [ -s Strict ]
             LogicalVolume Partitions [ PhysicalVolume ... ]
```

Description

Uses C-SPOC to run the **extendlv** command with the specified parameters, and make the updated logical volume definition available on all cluster nodes.

Flags

-a Position
Sets the physical volume allocation policy (the position of the logical partitions on the physical volume). The following *Position* variables are valid:

m Allocates logical partitions in the outer middle section of each physical volume. This variable is the default setting.

c Allocates logical partitions in the center section of each physical volume.

e Allocates logical partitions in the outer edge section of each physical volume.

ie Allocates logical partitions in the inner edge section of each physical volume.

im Allocates logical partitions in the inner middle section of each physical volume.

-e Range
Sets the physical volume allocation policy. The allocation policy is the number of physical volumes to extend across by using the volumes that provide the best allocation. The value of the *Range* variable is limited by the *Upperbound* variable that is set with the **-u** flag. The following *Range* variables are valid:

x Allocates logical partitions across the maximum number of physical volumes.

m Allocates logical partitions across the minimum number of physical volumes.

-s Strict
Determines the strict allocation policy. You can allocate copies of a logical partition to be shared or not shared for the same physical volume. The following *Strict* variables are valid:

y Sets a strict allocation policy, so copies of a logical partition cannot share the same physical volume.

n Does not set a strict allocation policy, so copies of a logical partition can share the same physical volume.

s Sets a super strict allocation policy, so that the partitions allocated for one mirror cannot share a physical volume with the partitions from another mirror. When you change to a non-super strict logical volume to a super strict logical volume, you must use the **-u** flag.

-u Upperbound
Sets the maximum number of physical volumes for the new allocation. The value of the *Upperbound* variable is between one and the total number of physical volumes. When you use super strictness, the upper bound indicates the maximum number of physical volumes that are allowed for each mirror copy. When you use striped logical volumes, the upper bound must be multiple of *Stripe_width* variable.

Example

To increase the size of the logical volume that is name *lv01* by three logical partitions, enter:

```
cli_extendlv lv01 3
```

Related information:

extendlv command

---

cli_extendvg command

Purpose

Adds physical volumes to a volume group on all nodes in a cluster.

Syntax

```
cli_extendvg VolumeGroup PhysicalVolume ...
```

Description

Uses C-SPOC to run the **extendvg** command with the specified parameters, and make the updated volume group definition available on all cluster nodes.

You must verify that the physical volumes (hdisks), that are going to be included, are available to all cluster nodes and have PVIDs assigned before you run this command.

Example

To add disks that are named *hdisk101* and *hdisk111* to a volume group that is named *vg01*, enter:

```
cli_extendvg vg01 hdisk101 hdisk111
```

Related information:

extendvg command

---

cli_importvg command

Purpose

Imports a new volume group definition from a set of physical volumes on all nodes in a cluster

Syntax

```
cli_importvg [ -y VolumeGroup ] [ -V MajorNumber ]  PhysicalVolume
```

Description

Uses C-SPOC to run the **importvg** command with the specified parameters. This command causes the Logical Volume Manager (LVM) on each cluster node to read the LVM information on the disks in the volume group, and update the local volume group definition.

Flags

-V MajorNumber
Specifies the major number of the imported volume group.

-y VolumeGroup
Specifies the name to use for the new volume group. If you do not use this flag, the system automatically generates a new name. The volume group name can contain only the following characters:

- A - Z
- a - z
- 0 - 9
- _ (underscore character)
- - (minus character)
- . (period character)

Example

To make the volume group that is name *bkvg* from the physical volume that is named *hdisk07* available on all cluster nodes, enter:

```
cli_importvg -y bkvg hdisk07
```

Related information:

importvg command

cli_mirrorvg command

Purpose

Syntax

```
cli_mirrorvg  [-S | -s] [-Q] [-c Copies] [-m] VolumeGroup [PhysicalVolume...]
```

Description

Uses C-SPOC to run the **mirrorvg** command with the specified parameters, and make the updated volume group definition available on all cluster nodes.

Flags

-c Copies
Specifies the minimum number of copies that each logical volume must have after you run the **mirrorvg** command. It might be possible, through the independent use of the **mklvcopy** command, that some logical volumes might have more than the minimum number specified after you run the **mirrorvg** command. The minimum value that you can specify is 2 and the maximum value you can specify is 3. A value of 1 is ignored.

-m exact map
Allows mirroring of logical volumes in the exact physical partition order that is in the original copy. You must specify a physical volume where the exact map copy is placed. If the space is insufficient for an exact mapping, then the command fails. You must add new drives or pick a different set of drives that satisfy an exact logical volume mapping of the entire volume group. The designated disks must be equal to or exceed the size of the drives that are being mirrored (regardless of if the entire disk is used). If any logical volume has already been mirrored the command fails.

-Q Quorum Keep
By default, when the content of a volume group is mirrored, the quorum for the volume group is disabled. If you want to keep the volume group quorum requirement after mirroring is complete, you can use this flag. For later quorum changes, see the **chvg** command.

-S Background Sync
Returns the **mirrorvg** command immediately and starts the **syncvg** command of the volume group in the a background. If you use this flag, it is not obvious when the mirrors complete their synchronization. However, as portions of the mirrors become synchronized, they are immediately used by the Logical Volume Manager (LVM) for mirroring.

-s Disable Sync
Returns the **mirrorvg** command immediately without performing any type of mirror synchronization. If you use this flag, the mirror might exist for a logical volume but it is not used by the operating system until it has been synchronized with the **syncvg** command.

Example

To specify two copies for every logical volume in shared volume group that is named *vg01*, enter:

```
cli_mirrorvg -c 2 vg01
```

Related information:
mirrorvg command

cli_mklv command

Purpose

Create a new logical volume on all nodes in a cluster.

Syntax

```
cli_mklv [ -a Position ] [ -b BadBlocks ] [ -c Copies ]  [ -d Schedule ]
         [ -e Range ] [ -i ] [ -L Label ] [ -o y / n ] [ -r Relocate ]
         [ -s Strict ] [ -t Type ] [ -u UpperBound ] [ -v Verify ]
         [ -w MirrorWriteConsistency ] [ -x Maximum ] [ -y NewLogicalVolume |
         -Y Prefix ] [ -S StripSize ] [ -U Userid ] [ -G Groupid ] [ -P Modes ]
   VolumeGroup NumberOfLPs [ PhysicalVolume ... ]
```

Description

Uses C-SPOC to run the **mklv** command with parameters, and make the new logical volume definition available on all cluster nodes.

Flags

-a Position
Sets the physical volume allocation policy (the position of the logical partitions on the physical volume). The following *Position* variables are valid:

- **m** Allocates logical partitions in the outer middle section of each physical volume. This variable is the default setting.
- **c** Allocates logical partitions in the center section of each physical volume.
- **e** Allocates logical partitions in the outer edge section of each physical volume.
- **ie** Allocates logical partitions in the inner edge section of each physical volume.
- **im** Allocates logical partitions in the inner middle section of each physical volume.

-b BadBlocks
Sets the bad-block relocation policy. The following *BadBlocks* variables are valid:

- **y** Causes bad-block relocation to occur.
- **n** Prevents bad block relocation from occurring.

-c Copies
Specifies the minimum number of copies that each logical volume must have after you run the **mirrorvg** command. It might be possible, through the independent use of the **mklvcopy** command, that some logical volumes might have more than the minimum number specified after you run the **mirrorvg** command. The minimum value that you can specify is 2 and the maximum value you can specify is 3. A value of 1 is ignored.

-d Schedule
Sets the scheduling policy when more than one logical partition is written. You must use parallel or sequential processing to mirror a striped logical volume. The following *Schedule* variables are valid:

- **p** Establishes a parallel scheduling policy.
- **ps** Parallel write with sequential read policy. All mirrors are written in parallel but always read from the first mirror if it is available.
- **pr** Parallel write and reads are done for all mirrors. This policy is similar to the parallel policy, except an attempt is made to spread the reads to the logical volume more evenly across all mirrors.
- **s** Establishes a sequential scheduling policy. Use this variable when you specify policy of parallel or sequential strictness (super strictness).

-e Range
Sets the physical volume allocation policy. The allocation policy is the number of physical volumes to extend across by using the volumes that provide the best allocation. The value of the *Range* variable is limited by the *Upperbound* variable that is set with the **-u** flag. The following *Range* variables are valid:

x Allocates logical partitions across the maximum number of physical volumes.

m Allocates logical partitions across the minimum number of physical volumes.

-G Groupid
Specifies group ID for the logical volume special file.

-L Label
Sets the logical volume label. The maximum size for this variable is 127 characters.

-n NewLogicalVolume
Changes the name of the logical volume that is specified by the *NewLogicalVolume* variable. Logical volume names must be unique system wide and can have a maximum of 15 characters.

-p Permission
Sets the access permission to read/write or read-only. The following *Permission* variables are valid:

w Sets the access permission to read/write.

r Sets the access permission to read-only. Mounting a JFS file system on a read-only logical volume is not supported.

-P Modes
Specifies permissions (file modes) for the logical volume special file.

-r Relocate
Specifies whether you want to allow or prevent the relocation of the logical volume during reorganization. The following *Relocate* variables are valid:

y Allows the logical volume to be relocated during reorganization. If the logical volume is striped, you cannot use the **chlv** command to change the relocation flag to `y`.

n Prevents the logical volume from being relocated during reorganization.

-s Strict
Determines the strict allocation policy. You can allocate copies of a logical partition to be shared or not shared for the same physical volume. The following *Strict* variables are valid:

y Sets a strict allocation policy, so copies of a logical partition cannot share the same physical volume.

n Does not set a strict allocation policy, so copies of a logical partition can share the same physical volume.

s Sets a super strict allocation policy, so that the partitions allocated for one mirror cannot share a physical volume with the partitions from another mirror. When you change to a non-super strict logical volume to a super strict logical volume, you must use the `-u` flag.

-S StripSize
Specifies the number of bytes per strip (the strip size that is multiplied by the number of disks in an array equals the stripe size). Valid values include 4K, 8K, 16K, 32K, 64K, 128K, 256K, 512K, 1M, 2M, 4M, 8M, 16M, 32M, 64M, and 128M. You cannot use the `-d`, `-e`, and `-s` flags when you are creating a striped logical volume with this flag.

-t Type
Sets the logical volume type. The following are the standard types:

- jfs (journaled file systems)
- jfslog (journaled file system logs)
- jfs2 (enhanced journaled file system)
- jfs2log (enhanced journaled file system logs)
- paging (paging spaces)

You can define other logical volume types with this flag. You cannot create a striped logical volume of type boot. The default value is jfs. If a logical volume is created with a type of jfslog or jfs2log, C-SPOC automatically runs the **logform** command so that it can be used.

-U Userid
Specifies user ID for the logical volume special file.

-u Upperbound
Sets the maximum number of physical volumes for the new allocation. The value of the *Upperbound* variable is between one and the total number of physical volumes. When you use super strictness, the upper bound indicates the maximum number of physical volumes that are allowed for each mirror copy. When you use striped logical volumes, the upper bound must be multiple of *Stripe_width* variable.

-v Verify
Sets the write-verify state for the logical volume. Causes all writes to the logical volume either to be verified with a follow-up read or not to be verified with a follow-up read. The following *Verify* variables are valid:

y All writes to the logical volume are verified with a follow-up read.

n All writes to the logical volume are not verified with a follow-up read.

-w MirrorWriteConsistency
The following *MirrorWriteConsistency* variables are valid:

y Turns on active mirror write consistency. This variable verifies data consistency on the mirrored copies of a logical volume during normal I/O processing.

p Turns on passive mirror write consistency. This variable verifies data consistency on the mirrored copies during volume group synchronization after a system interruption. This function is only available on Big Volume Groups.

n No mirror write consistency.

-x Maximum
Sets the maximum number of logical partitions that can be allocated to the logical volume. The maximum number of logical partitions per logical volume is 32,512.

-y NewLogicalVolume
Specifies the logical volume name to use instead of using a system-generated name. Logical volume names must be unique system-wide name, and can range from 1-15 characters. The new name must be unique across all nodes on which the volume group is defined. The name cannot begin with a prefix already defined in the predefined device database (PdDv) class in the device configuration database for other devices.

-Y Prefix
Specifies the prefix value to use instead of the prefix in a system-generated name for the new logical volume. The prefix value must be less than or equal to 13 characters. The name cannot begin with a prefix already defined in the predefined device database (PdDv) class in the device configuration database for other devices, and it cannot be a name that is already used by another device.

Example

To make a logical volume in volume group that is named *vg02* with one logical partition and a total of two copies of the data, enter:

```
cli_mklv -c 2 vg01 1
```

Related information:

cli_mklvcopy command

Purpose

Increase the number of copies in each logical partition in a logical volume on all nodes in a cluster.

Syntax

```
cli_mklvcopy  [ -a Position ] [ -e Range ] [ -k ] [ -s Strict ]
  [ -u UpperBound ] LogicalVolume Copies [ PhysicalVolume... ]
```

Description

Uses C-SPOC to run the **mklvcopy** command with parameters, and make the updated logical volume definition available on all cluster nodes.

Flags

-a Position
Sets the physical volume allocation policy (the position of the logical partitions on the physical volume). The following *Position* variables are valid:

- **m** Allocates logical partitions in the outer middle section of each physical volume. This variable is the default setting.
- **c** Allocates logical partitions in the center section of each physical volume.
- **e** Allocates logical partitions in the outer edge section of each physical volume.
- **ie** Allocates logical partitions in the inner edge section of each physical volume.
- **im** Allocates logical partitions in the inner middle section of each physical volume.

-e Range
Sets the physical volume allocation policy. The allocation policy is the number of physical volumes to extend across by using the volumes that provide the best allocation. The value of the *Range* variable is limited by the *Upperbound* variable that is set with the **-u** flag. The following *Range* variables are valid:

- **x** Allocates logical partitions across the maximum number of physical volumes.
- **m** Allocates logical partitions across the minimum number of physical volumes.

-k Synchronizes data in the new partitions.

-s Strict
Determines the strict allocation policy. You can allocate copies of a logical partition to be shared or not shared for the same physical volume. The following *Strict* variables are valid:

- **y** Sets a strict allocation policy, so copies of a logical partition cannot share the same physical volume.
- **n** Does not set a strict allocation policy, so copies of a logical partition can share the same physical volume.
- **s** Sets a super strict allocation policy, so that the partitions allocated for one mirror cannot share a physical volume with the partitions from another mirror. When you change to a non-super strict logical volume to a super strict logical volume, you must use the **-u** flag.

-u Upperbound
Sets the maximum number of physical volumes for the new allocation. The value of the *Upperbound* variable is between one and the total number of physical volumes. When you use super strictness,

the upper bound indicates the maximum number of physical volumes that are allowed for each mirror copy. When you use striped logical volumes, the upper bound must be multiple of *Stripe_width* variable.

Example

To add physical partitions to the logical partitions of a logical volume that is named *lv01*so that total of three copies exist for each logical partition, enter:

```
cli_mklvcopy lv01 3
```

Related information:

mklvcopy command

---

cli_mkvg command

Purpose

Create a volume group on all nodes in a cluster.

Syntax

```
cli_mkvg   [ -B ] [ -P Partitions ] [ -t factor ] [ -C ] [ -G ] [ -x ] [ -s Size ]
     [ -V MajorNumber ] [ -v LogicalVolumes ] [ -y VolumeGroup ]
     PhysicalVolume ...
```

Description

You can use C-SPOC to run the **mkvg** command with parameters, and make the new logical volume definition available on all cluster nodes.

Flags

-B Creates a big-type volume group. This type of volume group can accommodate up to 128 physical volumes and 512 logical volumes. Because the vgda space was increased substantially, every vgda update operation (creating a logical volume, changing a logical volume, and adding a physical volume) might take considerably longer to run.

-C Creates an enhanced concurrent capable volume group. You can use this flag only in a configured PowerHA SystemMirror cluster. You can use this flag to create an enhanced concurrent capable volume group.

Enhanced concurrent volume groups use group services. Group services are available with PowerHA SystemMirror and must be configured before you activate a volume group in this mode.

Only enhanced concurrent capable volume groups are supported by a 64-bit kernel. Concurrent capable volume groups are not supported by a 64-bit kernel.

-P Partitions
Specifies the total number of partitions in the volume group. The *Partitions* variable is represented in units of 1024 partitions. The following values are valid for this flag:

- 32
- 64
- 128
- 256
- 512
- 768
- 1024
- 2048

The default value is 32 k (32768 partitions). You can use the **chvg** command to increase the number of partitions up to the maximum of 2048 k (2097152 partitions). This flag is only valid with the **-s** flag.

-s size
Sets the number of megabytes (MB) in each physical partition. The size variable is expressed in units of megabytes from 1 (1 MB) - 131072 (128 GB). The size variable must be equal to a power of 2 (example 1, 2, 4, 8). The default value for 32 physical volume groups and 128 physical volume groups is the lowest value that remains within the limitation of 1016 physical partitions per physical volume. The default value for scalable volume groups is the lowest value to accommodate 2040 physical partitions per physical volume.

-t factor
Changes the limit of the number of physical partitions per physical volume, which is specified by a factor. The factor must be 1 - 16 for 32 physical volumes groups. The factor must be 1 - 64 for 128 physical volume groups. The maximum number of physical partitions per physical volume for this volume group changes to a factor of x 1016. The default is the lowest value to remain within the physical partition limit of factor x 1016. The maximum number of physical volumes that can be included in the volume group is maxpvs/factor. This flag is ignored if you use the **-s** flag.

-V majornumber
Specifies the major number of the volume group that is created.

-v Specifies the number of logical volumes that can be created. The following values are valid for this flag:

- 256
- 512
- 1024
- 2048
- 4096

The default value is 256. You can use the **chvg** command to increase the number of logical volumes up to the maximum of 4096. This flag is only valid with the **-s** flag. The last logical volume is reserved for metadata.

-y volumegroup
Specifies the volume group name rather than having the name generated automatically. The volume group names must be unique system wide and can range 1-15 characters. The name cannot begin with a prefix already defined in the predefined device database (PdDv) class in the device configuration database for other devices. The volume group name that is created is sent to standard output. The volume group name can contain only the following characters:

- a - z
- 0 - 9
- _ (underscore character)
- - (minus character)
- . (period character)

Example

To create a volume group that contains disks named *hdisk3*, *hdisk5*, and *hdisk6* with a physical partition size set to 1 megabyte, enter:

```
cli_mkvg -s 1 hdisk3 hdisk5 hdisk6
```

Related information:

mkvg command

cli_on_cluster command

Purpose

Run a command on all nodes in the cluster.

Syntax

```
cli_on_cluster  [ -S | -P ]  'command string'
```

Description

Runs a command as root, either serially or in parallel, on all cluster nodes. The output from the command (stdout and stderr) is displayed at the command line. Each line of output is preceded by the node name followed by a colon.

Flags

-S Runs one command at a time on each node in the cluster. Once the command finishes, the next command is run.

-P Run the command in parallel on all nodes in the cluster simultaneously.

Example

To reboot every node in the cluster, enter:

```
cli_on_cluster -S 'shutdown -Fr'
```

cli_on_node command

Purpose

Run an arbitrary command on a specific node in the cluster.

Syntax

```
cli_on_node  [ -V <volume group> | -R <resource group | -N <node> ] 'command string'
```

Description

Runs a command as root on either an explicitly specified node, or on the cluster node that owns a specified volume group or resource group. Any output from the command (stdout and stderr) is displayed at the command line.

Flags

-V volume group
Runs the command on the node or nodes on which the specified volume group is in a varied on state. If the volume group is in a varied on state in concurrent mode on multiple nodes, the command is run on all nodes.

-R resource group
Runs the command on the node that currently owns the specified resource group.

-N node
Runs the command on the specified node. This flag identifies the PowerHA SystemMirror node name.

Example

To run the **ps -efk** command on the node that is named awesome, enter:

```
cli_on_node -N awesome 'ps -efk'
```

cli_reducevg command

Purpose

Removes a physical volume from a volume group and make the updated changes available on all cluster nodes. When all physical volumes are removed from the volume group, the volume group is deleted on all cluster nodes.

Syntax

```
cli_reducevg  VolumeGroup PhysicalVolume ...
```

Description

Uses C-SPOC to run the **reducevg** command with parameters, and make the updated volume group definition available on all cluster nodes.

Example

To remove physical disk that is named *hdisk10* from a volume group that is named *vg01*, enter:

```
cli_reducevg vg01 hdisk101
```

Related information:

reducevg command

cli_replacepv command

Purpose

Replace a physical volume in a volume group with another physical volume and make the changes available on all cluster nodes.

Syntax

```
cli_replacepv  SourcePhysicalVolume  DestinationPhysicalVolume
```

Description

Uses C-SPOC to run the **replacepv** command with parameters, and make the updated volume group definition available on all cluster nodes.

Example

To replace a disk that is named *hdisk10* with a disk that is named *hdisk20* in the volume group that owns the *hdisk10* disk, enter:

```
cli_replacepv hdisk10 hdisk20
```

Related information:

replacepv command

cli_rmfs command

Purpose

Remove a file system from all nodes in a cluster.

Syntax

```
cli_rmfs [ -r ] FileSystem
```

Description

Uses C-SPOC to run the **rmfs** command with parameters, and remove the file system definition from all cluster nodes.

Flags

-r Removes the mount point of the file system

Example

To remove the shared file system called */test_fs*, enter:

```
cli_rmfs -r /test_fs
```

Related information:

rmfs command

cli_rmlv command

Purpose

Remove a logical volume from all nodes in a cluster.

Syntax

```
cli_rmlv  LogicalVolume ...
```

Description

Uses C-SPOC to run the **rmlv** command with parameters, and make the updated logical volume definition available on all cluster nodes.

Example

To change the shared logical volume that is named *lv01*, enter:

```
cli_rmlv lv01
```

Related information:

rmlv command

cli_rmlvcopy command

Purpose

Remove copies from a logical volume on all nodes in a cluster.

Syntax

```
cli_rmlvcopy LogicalVolume Copies [ PhysicalVolume... ]
```

Description

Uses C-SPOC to run the **rmlvcopy** command with parameters on all cluster nodes.

Example

To reduce the number of copies of each logical partition that belong to a logical volume that is named *lv01* so that there is only a single copy, enter:

```
cli_rmlvcopy lv01 1
```

Related information:

rmlvcopy command

---

cli_syncvg command

Purpose

Run the **syncvg** command with parameters and make the updated volume group definition available on all cluster nodes.

Syntax

```
cli_syncvg  [-f] [-H] [-P NumParallelLps] {-l|-v} Name
```

Description

Uses C-SPOC to run the **syncvg** command, which causes the Logical Volume Manager (LVM) for each cluster node to read the LVM information on the disks in the volume group. This command also updates the local volume group definition.

Flags

-f Specifies a good physical copy is selected and propagated to all other copies of the logical partition, even if they are not stale.

-H Delays the write operation, until the sync operation completes, for the selected volume group on any other cluster nodes where the concurrent volume group is active. When you use this flag, it is not required that all nodes in the cluster support the **-P** flag for the **cli_syncvg** command. This flag is ignored if the volume group is not varied on in concurrent mode.

-l Specifies that the *Name* variable represents a logical volume device name.

-P NumParallelLps
Specifies the number of logical partitions that are synchronized in parallel. The valid range for the *NumParallelLps* variable is 1 - 32. The *NumParallelLps* variable must be specific to the system, disks in the volume group, system resources, and volume group mode.

-v Specifies that the *Name* variable represents a volume group device name.

Example

To synchronize the copies on a volume group that is named *v01*, enter:

```
cli_syncvg -v vg01
```

Related information:

syncvg command

cli_unmirrorvg command

Purpose

Unmirror a volume group on all nodes in a cluster.

Syntax

```
cli_unmirrorvg  [ -c Copies ] VolumeGroup [ PhysicalVolume ... ]
```

Description

Uses C-SPOC to run the **unmirrorvg** command with parameters, and make the updated volume group definition available on all cluster nodes.

Flags

-c Copies
Specifies the minimum number of copies that each logical volume must have after you run the **unmirrorvg** command. If you do not want all logical volumes to have the same number of copies, then reduce the mirrors manually with the **rmlvcopy** command. If you do not use this flag, the default value of 1 is used.

Example

To specify a single copy for shared volume group that is named *vg01*, enter:

```
cli_unmirrorvg -c 1 vg01
```

Related information:

unmirrorvg command

cli_updatevg command

Purpose

Updates the definition of a volume group on all cluster nodes to match the current actual state of the volume group.

Syntax

```
cli_updatevg VolumeGroup
```

Description

Uses C-SPOC to run the **updatevg** command, which causes the Logical Volume Manager (LVM) on each cluster node to read the LVM information on the disks in the volume group and update the local volume group definition.

Example

To update the volume group definition for volume group that is name *vg11* on all cluster nodes, enter:

```
cli_updatevg vg11
```

cllscf command

Purpose

List cluster topology information.

Syntax

```
cllscf
```

Description

The **cllscf** command lists the cluster topology information that is defined in the cluster, network, and adapter configuration ODM object classes. The **cllscf** command summarizes the cluster configuration information.

Examples

To display cluster information defined in the default or active cluster configuration, enter:

```
cllscf
```

The command displays output information similar to the following example:

```
# /usr/es/sbin/cluster/utilities/cllscf
Cluster Name:    hadev11_cluster
Cluster Type:    Standard
Heartbeat Type:  Unicast
Repository Disk: hdisk10 (00c0f592e54367f2)

There were 2 networks defined: net_ether_01, net_ether_02
There are 2 nodes in this cluster

NODE hadev11:
        This node has 0 service IP label(s):

NODE hadev12:
        This node has 0 service IP label(s):

Breakdown of network connections:

Connections to network net_ether_01
        Node hadev11 is connected to network net_ether_01 by these interfaces:
                hadev11

        Node hadev12 is connected to network net_ether_01 by these interfaces:
                hadev12

Connections to network net_ether_02
        Node hadev12 is connected to network net_ether_02 by these interfaces:
                hadev12_en1_boot
                hadev12_en2_boot
```

Related reference:

"clmgr command" on page 40

cllsdisk command

Purpose

Lists PVIDs of accessible disks in a specified resource chain.

Syntax

```
cllsdisk {-g Resource Group }
```

Example

Run the following command to list PVIDs of disks accessible by all participating nodes in resource group *grp3*.

cllsdisk -g grp3

cllsfs command

Purpose

Lists shared file systems accessible by all participating nodes in a resource group.

Syntax

```
cllsfs {-g resource group } [-n]
```

Table 3. cllsfs flags

Flag	Description
-g resource group	Specifies name of resource group for which to list file systems.
-n	Lists the nodes that share the file system in the resource group.

Note: Do not run the **cllsfs** command from the command line. Use the SMIT interface to retrieve file system information, as explained in (See Managing Shared LVM Components).

cllsgrp command

Purpose

Lists all resource groups that are configured for a cluster.

Syntax

```
cllsgrp
```

Description

Displays the names of all resource groups in the cluster.

Example

To display resource group information for a cluster, enter:

```
cllsgrp
```

The command displays the following output:

```
grp1
grp2
grp3
grp4
```

cllsparam command

Purpose

Lists runtime parameters.

Syntax

```
cllsparam {-n nodename } [-c] [-s] [-d odmdir ]
```

Flags

-n nodename
Specifies a node for which to list the information.

-c Specifies a colon output format.

-s Used along with the **-c** flag, specifies native language instead of English.

-d odmdir
Specifies an alternate ODM directory.

Example

Run the following example to display runtime parameters for node abalone:

```
cllsparam -n abalone
```

---

cllsres command

Purpose

ISorts PowerHA SystemMirror for AIX Configuration Database resource data by name and arguments.

Syntax

```
cllsres [-g group ] [-e] [-c] [-s] [-d odmdir ] [-qquery ]
```

Flags

-g group
Specifies name of resource group to list.

-c Specifies a colon output format.

-e Expands the user-defined resource list in "resourcetype=resourcenames" format.

-s Used with the **-c** flag, specifies native language instead of English.

-d odmdir
Specifies an alternate ODM directory.

-q query
Specifies search criteria for ODM retrieve. See the **odmget** man page for information on search criteria.

Examples

1. Run the following command to list source data for all resource groups.
   ```
   cllsres
   ```
2. Run the following command to list resource data for grp1 resource group.
   ```
   cllsres -g grp1
   ```
3. Run the following command to lists file system resource data for grp1 resource group.
   ```
   cllsres -g grp1 -q"name = FILESYSTEM"
   ```

---

cllsserv command

Purpose

Lists application controllers by name.

Syntax

```
cllsserv [-c] [-h] [-n name ] [-d odmdir ]
```

Flags

-c Specifies a colon output format.

-h Specifies to print a header.

-n name
Specifies an application controller for which to check information.

-d odmdir
Specifies an alternate ODM directory.

Examples

1. Run the following command to lists all application controllers.
   ```
   cllsserv
   ```
2. Run the following command to lists information in colon format for test1 application controller.
   ```
   cllsres -c -n test1
   ```

---

cllsvg command

Purpose

Lists volume groups shared by nodes in a cluster. A volume group is considered shared if it is accessible by all participating nodes in a configured resource group. Note that the volume groups listed may or may *not* be configured as a resource in any resource group. If neither **-s** nor **-c** is selected, then both shared and concurrent volume groups are listed.

Syntax

```
cllsvg {-g resource group } [-n] [-v] [-s | -c]]
```

Flags

-g resource group
Specifies name of resource group for which to list volume groups that are shared amongst nodes participating in that resource group.

-n nodes
Specifies all nodes participating in each resource group.

-v Lists only volume groups that are varied on, and match other command line criteria.

-s Lists only shared volume groups that also match other criteria.

-c Lists only concurrent volume groups that also match other criteria.

Example

Run the following command to list all shared volume groups in grp1 resource group.

```
cllsvg -g grp1
```

---

clmgr command

Purpose

The **clmgr** command provides a consistent, reliable interface for performing PowerHA SystemMirror cluster operations by using a terminal or script.

Syntax

The following is the full syntax for the **clmgr** command:

```
clmgr {[-c|-d <DELIMITER>] [-S] | [-x]}
[-v][-f][-D] [-T <#####>]
[-l {error|standard|low|med|high|max}][-a {<ATTR#1>,<ATTR#2>,...}] <ACTION> <CLASS> [<NAME>]
[-h | <ATTR#1>=<VALUE#1> <ATTR#2>=<VALUE#2> <ATTR#n>=<VALUE#n>]
```

```
clmgr {[-c|-d <DELIMITER>] [-S] | [-x]}
[-v][-f] [-D] [-T <#####>]
[-l {error|standard|low|med|high|max}][-a {<ATTR#1>,<ATTR#2>,...}]
[-M]- "
<ACTION> <CLASS> [<NAME>] <ATTR#1>=<VALUE#1> <ATTR#n>=<VALUE#n>]
.
.
."
         ACTION={add|modify|delete|query|online|offline|...}
          CLASS={cluster|site|node|network|resource_group|...}

    clmgr {-h|-?} [-v]
    clmgr [-v] help
```

The following is the basic format for using the **clmgr** command:

```
clmgr <ACTION> <CLASS> [<NAME>] [<ATTRIBUTES...>]
```

Help is available for the **clmgr** command from the command line. For example, when you run the **clmgr** command without any flags or parameters a list of the available ACTIONs is displayed. Entering `clmgr ACTION` from the command line with no CLASS provided, results in a list of all the available CLASSes for the specified ACTION. Entering `clmgr ACTION CLASS` with no NAME or ATTRIBUTES provided is slightly different, because some ACTION+CLASS combinations do not require any additional parameters. To display help in this scenario, you must explicitly request the help by appending the `-h` flag to the `clmgr ACTION CLASS` command. You cannot display help from the command line for each of the **clmgr** commands individual ATTRIBUTES.

Description

The high degree of consistency used by the **clmgr** command helps make it easier to learn and use. In addition to consistency of execution, **clmgr** also provides consistent return codes to make scripting easier. Several output formats are also provided for data queries to make collecting cluster information as easy as possible.

All **clmgr** command operations are logged in the clutils.log file, including the name of the command that was executed, the commands start and stop time, and the user name that initiated the command.

Note: If the resource groups have more than one dependency, you cannot use the **clmgr** command to move multiple resource groups.

Flags

ACTION

Describes the operation to be performed.

Note: ACTION is not case-sensitive. All ACTION flags provide a shorter alias. For example, `rm` is an alias for `delete`. Aliases are provided for convenience from the command line and must not be used in scripts.

The following four ACTION flags are available on almost all the supported CLASS objects:

- add (Alias: a)
- query (Aliases: q, ls, get)

- modify (Aliases: mod, ch, set)
- delete (Aliases: de, rm, er)

The remaining ACTIONS are typically only supported on a small subset of the supported CLASS objects:

- Cluster, Node, Resource Group:
  - start (Aliases: online, on)
  - stop (Aliases: offline, off)
- Resource Group, Service IP, Persistent IP:
  - move (Alias: mv)
- Cluster, Interface, Log, Node, Snapshot, Network, Application Monitor:
  - manage (Alias: mg)
- Cluster and File Collection:
  - sync (Alias: sy)
- Cluster, Method:
  - verify (Alias: ve)
- Log, Report, Snapshot:
  - view (Alias: vi)
- Repository:
  - replace (Alias: rep, switch, swap)

CLASS

The type of object upon which the ACTION is performed.

Note: CLASS is not case-sensitive. All CLASS objects provide a shorter alias. For example, `fc` is an alias for `file_collection`. Aliases are provided for convenience from the command line and must not be used in scripts.

The following is the complete list of supported CLASS objects:

- cluster (Alias: cl)
- repository (Alias: rp)
- site (Alias: st)
- node (Alias: no)
- interface (Aliases: in, if)
- network (Aliases: ne, nw)
- resource_group (Alias: rg)
- service_ip (Alias: si)
- persistent_ip (Alias: pi)
- application_controller (Aliases: ac, app)
- application_monitor (Aliases: am, mon)
- tape (Alias: tp)
- dependency (Alias: de)
- file_collection (Aliases: fi, fc)
- snapshot (Aliases: sn, ss)
- method (Alias: me)
- volume_group (Alias: vg)
- logical_volume (Alias: lv)
- file_system (Alias: fs)

- physical_volume (Aliases: pv, disk)
- mirror_pool (Alias: mp)
- user (Alias: ur)
- group (Alias: gp)
- ldap_server (Alias: ls)
- ldap_client (Alias: lc)
- event
- hmc
- cod (Alias: cuod, dlpar)

Name
The specific object, of type CLASS, upon which the ACTION is to be performed.

ATTR=VALUE
An optional flag that has attribute pairs and value pairs that are specific to the ACTION+CLASS combination. Use these pairs flag to specify configuration settings or to adjust particular operations.

When used with the query action, the ATTR=VALUE specifications can be used to perform attribute-based searching and filtering. When used for this purpose, you can use simple wildcards. For example, "*" matches zero or greater of any character, "?" matches zero or one of any character.

Note: An ATTR might not always need to be fully typed out. Only the number of leading characters required to uniquely identify the attribute from the set of attributes available for the specified operation must be provided. Instead of entering `FC_SYNC_INTERVAL`, for the add cluster operation, you can enter `FC` for the same result.

-a Displays only the specified attributes, and is only valid with the query, add, and modify ACTIONs. Attribute names are not case-sensitive, and can be used with the standard UNIX wildcards, "*", and "?".

-c Displays all data in colon-delimited format, and is only valid with the query, add, and modify ACTIONs.

-d Valid only with the *query*, *add*, and *modify* ACTION flags, requests all data to be displayed in delimited format, using the specified delimiter.

-D Disables the dependency mechanism in **clmgr** command that attempts to create any requisite resources by using default values if they are not already defined within the cluster.

-f Overrides any interactive prompts, forcing the current operation to be attempted (if forcing the operation is a possibility).

-h Displays help information.

-l Activates the following trace logging values for serviceability:
- Error: Only updates the log file if an error is detected.
- Standard: Logs basic information for every `clmgr` operation.
- Low: Basic entry and exit tracing for every function.
- Med: Performs *low* tracing, adding function entry parameters, and function return values.
- High: Performs *med* tracing, adding tracing of every line of execution, omitting routine, utility functions.
- Max - Performs *high* tracing, adding the routine function and utility function. Adds a time and date stamp to the function entry message and exit message.

Note: All trace data is written into the clutils.log file. This flag is ideal for troubleshooting problems.

-M
Allows multiple operations to be specified and run via one invocation of **clmgr**, with one operation being specified per line. All the operations will share a common transaction ID.

-S Displays data with column headers suppressed, and is only valid with the query ACTION and -c flag.

-T A transaction ID is applied to all logged output, to help group one or more activities into a single body of output that can be extracted from the log for analysis. This flag is ideal for troubleshooting problems.

-v Displays maximum verbosity in the output.

Note: Displays all instances of the specified class, when used with the query ACTION and no specific object name. For example, entering `clmgr -v query node` display all nodes and their attributes. Displays resulting attributes after the operation is complete (only if the operation was successful), when this flag is used with the add or modify ACTION.

-x Displays all data in a simple XML format, and is only valid with the query, add, and modify ACTIONs.

Syntax

The following sections describe the syntax for all possible **clmgr** operations.

- Application controller
- Application monitor
- Cluster
- CoD
- Dependency
- EFS
- Event
- Fallback timer
- File collection
- File system
- Group
- HMC
- Interface
- LDAP Server
- LDAP Client
- Log
- Method
- Mirror group
- Mirror pair
- Mirror pool
- Network
- Node
- Persistent IP/Label
- Physical volume
- Report
- Repository
- Resource group

- Service IP/Label
- Site
- Snapshot
- Storage agent
- Storage system
- Tape
- User
- Volume group

Cluster

```
clmgr add cluster \
      [ <cluster_label> ] \
      [ NODES=<host>[,<host#2>,...] ] \
      [ TYPE={NSC|SC} ] \
      [ HEARTBEAT_TYPE={unicast|multicast} ] \
          [ CLUSTER_IP=<IP_Address> ] \
      [ REPOSITORIES=<disk>[,<backup_disk>,...] ] \
      [ FC_SYNC_INTERVAL=## ] \
      [ RG_SETTLING_TIME=## ] \
      [ MAX_EVENT_TIME=### ] \
      [ MAX_RG_PROCESSING_TIME=### ] \
      [ DAILY_VERIFICATION={Enabled|Disabled} ] \
      [ VERIFICATION_NODE={Default|<node>} ] \
      [ VERIFICATION_HOUR=<00..23> ] \
      [ VERIFICATION_DEBUGGING={Enabled|Disabled} ] \
      [ HEARTBEAT_FREQUENCY=<min..max> ] \
      [ GRACE_PERIOD=<min..max> ] \
      [ SITE_POLICY_FAILURE_ACTION={fallover|notify} ] \
      [ SITE_POLICY_NOTIFY_METHOD="<FULL_PATH_TO_FILE>" ] \
      [ SITE_HEARTBEAT_CYCLE=<min..max> ] \
      [ SITE_GRACE_PERIOD=<min..max> ] \
      [ TEMP_HOSTNAME={disallow|allow} ] \
      [ MONITOR_INTERFACES={enable|disable} ] \
      [ NETWORK_FAILURE_DETECTION_TIME=<0..590> ]

clmgr add cluster \
      [ <cluster_label> ] \
      [ NODES=<host>[,<host#2>,...] ] \
      [ TYPE="LC" ] \
      [ HEARTBEAT_TYPE={unicast|multicast} ] \
      [ FC_SYNC_INTERVAL=## ] \
      [ RG_SETTLING_TIME=## ] \
      [ MAX_EVENT_TIME=### ] \
      [ MAX_RG_PROCESSING_TIME=### ] \
      [ DAILY_VERIFICATION={Enabled|Disabled} ] \
      [ VERIFICATION_NODE={Default|<node>} ] \
      [ VERIFICATION_HOUR=<00..23> ] \
      [ VERIFICATION_DEBUGGING={Enabled|Disabled} ] \
      [ HEARTBEAT_FREQUENCY=<min..max> ] \
      [ GRACE_PERIOD=<min..max> ] \
      [ SITE_POLICY_FAILURE_ACTION={fallover|notify} ] \
      [ SITE_POLICY_NOTIFY_METHOD="<FULL_PATH_TO_FILE>" ]
      [ SITE_HEARTBEAT_CYCLE=<min..max> ] \
      [ SITE_GRACE_PERIOD=<min..max> ] \
      [ TEMP_HOSTNAME={disallow|allow} ] \
      [ MONITOR_INTERFACES={enable|disable} ] \
      [ NETWORK_FAILURE_DETECTION_TIME=<0..590> ]
```

Table 4. Acronyms and their meaning

Acronym	Meaning
NSC	Nonsite cluster (no sites will be defined)
SC	Stretched cluster (simplified infrastructure, ideal for limited distance data replication; sites must be defined)
LC	Linked cluster (full-featured infrastructure, ideal for long distance data replication; sites must be defined).

Note: *CLUSTER_IP* can only be used with a cluster type of *NSC* or *SC*. For *LC* clusters, the multicast address must be set for each site.

Note: The *REPOSITORIES* option can be used only with a cluster type of *NSC* or *SC*. For *LC* clusters, the *REPOSITORIES* option is identified for each site. The *REPOSITORIES* option can use seven disks. The first disk is the active repository disk and the following disks are the backup repositories disks.

```
clmgr modify cluster \
      [ NAME=<new_cluster_label> ] \
      [ NODES=<host>[,<host#2>,...] ] \
      [ TYPE={NSC|SC} ] \
      [ HEARTBEAT_TYPE={unicast|multicast} ] \
      [ CLUSTER_IP=<IP_Address> ] \
      [ REPOSITORIES=<backup_disk>[,<backup_disk>,...] ] \
      [ FC_SYNC_INTERVAL=## ] \
      [ RG_SETTLING_TIME=## ] \
      [ MAX_EVENT_TIME=### ] \
      [ MAX_RG_PROCESSING_TIME=### ] \
      [ DAILY_VERIFICATION={Enabled|Disabled} ] \
      [ VERIFICATION_NODE={Default|<node>} ] \
      [ VERIFICATION_HOUR=<00..23> ] \
      [ VERIFICATION_DEBUGGING={Enabled|Disabled} ] \
      [ HEARTBEAT_FREQUENCY=<min..max> ] \
      [ GRACE_PERIOD=<min..max> ] \
      [ SITE_POLICY_FAILURE_ACTION={fallover|notify} ] \
      [ SITE_POLICY_NOTIFY_METHOD="<FULL_PATH_TO_FILE>" ] \
      [ SITE_HEARTBEAT_CYCLE=<min..max> ] \
      [ SITE_GRACE_PERIOD=<min..max> ] \
      [ TEMP_HOSTNAME={disallow|allow} ] \
      [ MONITOR_INTERFACES={enable|disable} ] \
      [ LPM_POLICY={manage|unmanage} ] \
      [ HEARTBEAT_FREQUENCY_DURING_LPM=### ] \
      [ NETWORK_FAILURE_DETECTION_TIME=<0..590> ]
```

Note: The *REPOSITORIES* option can be used only with a cluster type of *NSC* or *SC*. For *LC* clusters, the *REPOSITORIES* option is identified for each site. The *REPOSITORIES* option can use six backup repository disks.

```
clmgr modify cluster \
      [ NAME=<new_cluster_label> ] \
      [ NODES=<host>[,<host#2>,...] ] \
      [ TYPE="LC" ] \
      [ HEARTBEAT_TYPE={unicast|multicast} ] \
      [ FC_SYNC_INTERVAL=## ] \
      [ RG_SETTLING_TIME=## ] \
      [ MAX_EVENT_TIME=### ] \
      [ MAX_RG_PROCESSING_TIME=### ] \
      [ DAILY_VERIFICATION={Enabled|Disabled} ] \
      [ VERIFICATION_NODE={Default|<node>} ] \
      [ VERIFICATION_HOUR=<00..23> ] \
      [ VERIFICATION_DEBUGGING={Enabled|Disabled} ] \
      [ HEARTBEAT_FREQUENCY=<min..max> ] \
      [ GRACE_PERIOD=<min..max> ] \
```

```
      [ SITE_POLICY_FAILURE_ACTION={fallover|notify} ] \
      [ SITE_POLICY_NOTIFY_METHOD="<FULL_PATH_TO_FILE>" ]
      [ SITE_HEARTBEAT_CYCLE=<min..max> ] \
      [ SITE_GRACE_PERIOD=<min..max> ] \
      [ TEMP_HOSTNAME={disallow|allow} ] \
      [ MONITOR_INTERFACES={enable|disable} ] \
      [ LPM_POLICY={manage|unmanage} ] \
      [ HEARTBEAT_FREQUENCY_DURING_LPM=### ] \
      [ NETWORK_FAILURE_DETECTION_TIME=<0..590> ]
clmgr modify cluster \
      [ SPLIT_POLICY={none|tiebreaker|manual|NFS} ] \
      [ TIEBREAKER=<disk> ] \
      [ MERGE_POLICY={none|majority|tiebreaker|manual|NFS} ] \
      [ NFS_QUORUM_SERVER=<server> ] \
      [ LOCAL_QUORUM_DIRECTORY=<local_mount>] \
      [ REMOTE_QUORUM_DIRECTORY=<remote_mount>] \
      [ QUARANTINE_POLICY=<disable|node_halt|fencing|halt_with_fencing>] \
      [ CRITICAL_RG=<rgname> ] \
      [ NOTIFY_METHOD=<method> ] \
      [ NOTIFY_INTERVAL=### ] \
      [ MAXIMUM_NOTIFICATIONS=### ] \
      [ DEFAULT_SURVIVING_SITE=<site> ] \
      [ APPLY_TO_PPRC_TAKEOVER={yes|no} ] \
      [ ACTION_PLAN={reboot|disable_rgs_autostart|disable_cluster_services_autostart} ]
```

Note: After sites are fully defined and synchronized and if sites are already in use, the cluster type cannot be modified.

```
clmgr query cluster [ ALL | {CORE,SECURITY,SPLIT-MERGE,HMC,ROHA} ]
clmgr delete cluster [ NODES={ALL|<node>[,<node#2>,...}] ]
```

Note: The delete action defaults to deleting the cluster completely, from all available nodes.

```
clmgr discover cluster
clmgr recover cluster
clmgr sync cluster \
      [ VERIFY={yes|no} ] \
      [ CHANGES_ONLY={no|yes} ] \
      [ DEFAULT_TESTS={yes|no} ] \
      [ METHODS=<method#1>[,<method#2>,...] ] \
      [ FIX={no|yes} ] \
      [ LOGGING={standard|verbose} ] \
      [ LOGFILE=<PATH_TO_LOG_FILE> ] \
      [ MAX_ERRORS=## ] \
      [ FORCE={no|yes} ]
```

Note: All options are verification parameters, so they are only valid when VERIFY is set to yes.

```
clmgr manage cluster {reset|unlock}

clmgr manage cluster security \
      [ LEVEL={Disable|Low|Med|High} ] \
      [ ALGORITHM={DES|3DES|AES} ]\
      [ GRACE_PERIOD=<SECONDS> ] \
      [ REFRESH=<SECONDS> ] ] \
      [ MECHANISM={OpenSSL|SSH} ] \
        CERTIFICATE=<PATH_TO_FILE> \
        PRIVATE_KEY=<PATH_TO_FILE>
```

Note: If a MECHANISM of SSL or SSH is specified, then a custom made certificate and private key file must be provided.

```
clmgr manage cluster security \
      [ LEVEL={Disable|Low|Med|High} ] \
```

```
      [ ALGORITHM={DES|3DES|AES} ]\
      [ GRACE_PERIOD=<SECONDS> ] \
      [ REFRESH=<SECONDS> ] ] \
      [ MECHANISM="SelfSigned" ] \
        [ CERTIFICATE=<PATH_TO_FILE> ] \
        [ PRIVATE_KEY=<PATH_TO_FILE> ]
```

Note: If a MECHANISM of Self-Signed is specified, then specifying a certificate and private key file is optional. If neither is provided, a default pair is automatically generated. GRACE_PERIOD defaults to 21600 seconds (6 hours). REFRESH defaults to 86400 seconds (24 hours).

```
clmgr manage cluster hmc \
            [ DEFAULT_HMC_TIMEOUT=<MINUTES> ] \
            [ DEFAULT_HMC_RETRY_COUNT=<INTEGER> ] \
            [ DEFAULT_HMC_RETRY_DELAY=<SECONDS> ] \
            [ DEFAULT_HMCS_LIST=<HMCS> ]

clmgr manage cluster roha \
            [ ALWAYS_START_RG={YES|NO} ] \
            [ ADJUST_SPP_SIZE={YES|NO} ]\
            [ FORCE_SYNC_RELEASE={YES|NO} ] \
            [ AGREE_TO_COD_COSTS={YES|NO} ] ] \
            [ ONOFF_DAYS=<DAYS>} ]
            [ RESOURCE_ALLOCATION_ORDER={enterprise_pool_first|free_pool_first} ]

clmgr verify cluster \
      [ CHANGES_ONLY={no|yes} ] \
      [ DEFAULT_TESTS={yes|no} ] \
      [ METHODS=<method#1>[,<method#2>,...] ] \
      [ FIX={no|yes} ] \
      [ LOGGING={standard|verbose} ] \
      [ LOGFILE=<PATH_TO_LOG_FILE> ] \
      [ MAX_ERRORS=## ]
      [ SYNC={no|yes} ] \
      [ FORCE={no|yes} ]
```

Note: The FORCE option can be used when SYNC is set to yes.

```
clmgr offline cluster \
      [ WHEN={now|restart|both} ] \
      [ MANAGE={offline|move|unmanage} ] \
      [ BROADCAST={true|false} ] \
      [ TIMEOUT=<seconds_to_wait_for_completion> ]
      [ STOP_CAA={no|yes} ]
clmgr online cluster \
      [ WHEN={now|restart|both} ] \
      [ MANAGE={auto|manual|delayed} ] \
      [ BROADCAST={false|true} ] \
      [ CLINFO={false|true|consistent} ] \
      [ FORCE={false|true} ] \
      [ FIX={no|yes|interactively} ] \
      [ TIMEOUT=<seconds_to_wait_for_completion> ] \
      [ START_CAA={no|yes} ]
```

Note: The RG_SETTLING_TIME attribute only affects resource groups with a startup policy of Online On First Available Node. An alias for cluster is cl.

Note: The STOP_CAA and START_CAA options bring the Cluster Aware AIX (CAA) cluster services offline or online. Use these options when there is a specific known need for them, or at the direction of IBM® support. Do not deactivate CAA cluster services because it disables the ability to detect problems in the clustered environment. The `only` option starts only CAA services.

Repository

```
clmgr add repository <disk>[,<backup_disk#2>,...] \
      [ SITE=<site_label> ]\
      [ NODE=<reference_node> ]
```

Note: If an active repository is not already defined, the first disk is used as the active repository. Any other disks in the list are defined as backup repository disks. You can identify up to six backup repository disks per cluster for standard clusters and stretched clusters. You can identify up to six backup repository disks per site for linked clusters.

```
clmgr replace repository [ <new_repository> ] \
      [ SITE=<site_label> ] \
      [ NODE=<reference_node>]
```

Note: If no disk is specified, the first disk in the backup list is used.

```
clmgr query repository [ <disk>[,<disk#2>,...] ]
clmgr delete repository {<backup_disk>[,<disk#2>,...] | ALL}\
      [ SITE=<site_label> ] \
      [ NODE=<reference_node> ]
```

Note: It is not possible to delete an active repository disk. Only backup repositories can be removed.

Site

```
clmgr add site <sitename> \
      NODES=<node>[,<node#2>,...] \
      [ SITE_IP=<multicast_address> ] \
      [ RECOVERY_PRIORITY={MANUAL|1|2} ] \
      [ REPOSITORIES=<disk>[,<backup_disk>,...] ]
```

Note: The *REPOSITORIES* option can be used only with a cluster type of *LC*. The *REPOSITORIES* option can use seven disks. The first disk is the active repository disk and the following disks are the backup repositories disks.

```
clmgr modify site <sitename> \
      [ NAME=<new_site_label> ] \
      [ NODES=<node>[,<node#2>,...] ] \
      [ SITE_IP=<multicast_address> ] \
      [ RECOVERY_PRIORITY={MANUAL|1|2} ] \
      [ REPOSITORIES=<backup_disk>[,<backup_disk>,...] ] \
      [ HMCS=<hmc>[,<hmc#2>,...] ]
```

Note: The *SITE_IP* attribute can be used only with a cluster type of *LC* (linked clusters) and a cluster heartbeat type of *multicast*.

Note: The *REPOSITORIES* option can be used only with a cluster type of *LC*. The *REPOSITORIES* option can use six backup repository disks.

```
clmgr query site [ <sitename>[,<sitename#2>,...] ]
clmgr delete site {<sitename>[,<sitename#2>,...] | ALL}
clmgr offline site <sitename> \
      [ WHEN={now|restart|both} ] \
      [ MANAGE={offline|move|unmanage} ] \
      [ BROADCAST={true|false} ] \
      [ TIMEOUT=<seconds_to_wait_for_completion> ] \
      [ STOP_CAA={no|yes} ]
clmgr online site <sitename> \
      [ WHEN={now|restart|both} ] \
      [ MANAGE={auto|manual} ] \
      [ BROADCAST={false|true} ] \
      [ CLINFO={false|true|consistent} ] \
```

```
      [ FORCE={false|true} ] \
      [ FIX={no|yes|interactively} ] \
      [ TIMEOUT=<seconds_to_wait_for_completion> ] \
      [ START_CAA={no|yes|only} ]
clmgr manage site respond {continue|recover}
```

Note: An alias for *site* is st.

Note: The STOP_CAA and START_CAA options bring the Cluster Aware AIX (CAA) cluster services offline or online. Use these options when there is a specific known need for them, or at the direction of IBM support. Do not deactivate CAA cluster services because it disables the ability to detect problems in the clustered environment. The `only` option starts only CAA services.

Node

```
clmgr add node <node> \
      [ COMMPATH=<ip_address_or_network-resolvable_name> ] \
      [ RUN_DISCOVERY={true|false} ] \
      [ PERSISTENT_IP=<IP> NETWORK=<network>
        {NETMASK=<255.255.255.0 | PREFIX=1..128} ] \
      [ START_ON_BOOT={false|true} ] \
      [ BROADCAST_ON_START={true|false} ] \
      [ CLINFO_ON_START={false|true|consistent} ] \
      [ VERIFY_ON_START={true|false} ] \
      [ SITE=<sitename> ]
clmgr modify node <node> \
      [ NAME=<new_node_label> ] \
      [ COMMPATH=<new_commpath> ] \
      [ PERSISTENT_IP=<IP> NETWORK=<network>
        {NETMASK=<255.255.255.0 | PREFIX=1..128} ] \
      [ START_ON_BOOT={false|true} ] \
      [ BROADCAST_ON_START={true|false} ] \
      [ CLINFO_ON_START={false|true|consistent} ] \
      [ VERIFY_ON_START={true|false} ] \
      [ HMCS=<hmc>[,<hmc#2>,...] ] \
      [ ENABLE_LIVE_UPDATE={true|false} ]
clmgr query node [ {<node>|LOCAL}[,<node#2>,...] ]
clmgr delete node {<node>[,<node#2>,...] | ALL}
clmgr manage node undo_changes
clmgr recover node <node>[,<node#2>,...]
clmgr online node <node>[,<node#2>,...] \
      [ WHEN={now|restart|both} ] \
      [ MANAGE={auto|manual} ] \
      [ BROADCAST={false|true} ] \
      [ CLINFO={false|true|consistent} ] \
      [ FORCE={false|true} ] \
      [ FIX={no|yes|interactively} ] \
      [ TIMEOUT=<seconds_to_wait_for_completion> ] \
      [ START_CAA={no|yes|only} ]
clmgr offline node <node>[,<node#2>,...] \
      [ WHEN={now|restart|both} ] \
      [ MANAGE={offline|move|unmanage} ] \
      [ BROADCAST={true|false} ] \
      [ TIMEOUT=<seconds_to_wait_for_completion> ] \
      [ STOP_CAA={no|yes} ]
```

Note: The TIMEOUT attribute defaults to 120 seconds. An alias for *node* is no.

Note: The STOP_CAA and START_CAA options bring the Cluster Aware AIX (CAA) cluster services offline or online. Use these options when there is a specific known need for them, or at the direction of IBM support. Do not deactivate CAA cluster services because it disables the ability to detect problems in the clustered environment. The `only` option starts only CAA services.

Network

```
clmgr add network <network> \
      [ TYPE={ether|XD_data|XD_ip} ] \
      [ {NETMASK=<255.255.255.0 | PREFIX=1..128} ] \
      [ IPALIASING={true|false} ] \
      [ PUBLIC={true|false} ]
```

Note: By default, an IPv4 network is constructed using a netmask of *255.255.255.0*. To create an IPv6 network, specify a valid prefix.

```
clmgr modify network <network> \
      [ NAME=<new_network_label> ] \
      [ TYPE={ether|XD_data|XD_ip} ] \
      [ {NETMASK=<255.255.255.0> | PREFIX=1..128} ] \
      [ PUBLIC={true|false} ] \
      [ RESOURCE_DIST_PREF={AC|ACS|C|CS|CPL|ACPL|ACPLS|NOALI} ] \
        [ SOURCE_IP=<service_or_persistent_ip> ]
```

Note: The possible values for the RESOURCE_DIST_PREF attribute follow:

AC Anti-collocation

ACS
Anti-collocation with source

C Collocation

CS Collocation with source

CPL
Collocation with persistent label

ACPL
Anti-collocation with persistent label

ACPLS
Anti-collocation with persistent label and source

NOALI
Disables first alias

Note: If the RESOURCE_DIST_PREF attribute uses the CS or ACS value, the SOURCE_IP attribute must be a service label.

```
clmgr query network [ <network>[,<network#2>,...] ]
clmgr delete network {<network>[,<network#2>,...] | ALL}
```

Note: Aliases for *network* are ne and nw.

Interface

```
clmgr add interface <interface> \
      NETWORK=<network> \
      [ NODE=<node> ] \
      [ TYPE={ether|XD_data|XD_ip} ] \
      [ INTERFACE=<network_interface> ]
clmgr modify interface <interface> \
      NETWORK=<network>
clmgr query interface [ <interface>[,<if#2>,...] ]
clmgr delete interface {<interface>[,<if#2>,...] | ALL}
clmgr discover interfaces
```

Note: The interface can be either an IP address or label. The NODE attribute defaults to the local node name. The TYPE attribute defaults to ether. The <network_interface> might look like en1, en2, en3. Aliases for *interface* are in and if.

Resource group

```
clmgr add resource_group <resource_group>[,<rg#2>,...] \
NODES=nodeA1,nodeA2,...                                 \
[ SECONDARYNODES=nodeB2[,nodeB1,...]                  ] \
[ SITE_POLICY={ignore|primary|either|both}            ] \
[ STARTUP={OHN|OFAN|OAAN|OUDP}                        ] \
[ FALLOVER={FNPN|FUDNP|BO}                            ] \
[ FALLBACK={NFB|FBHPN}                                ] \
  [ FALLBACK AT=<FALLBACK TIMER>                      ] \
[ NODE_PRIORITY_POLICY={default|mem|cpu|                \
                        disk|least|most}              ] \
[ NODE_PRIORITY_POLICY_SCRIPT=</path/to/script>       ] \
[ NODE_PRIORITY_POLICY_TIMEOUT=###                    ] \
[ SERVICE_LABEL=service_ip#1[,service_ip#2,...]       ] \
[ APPLICATIONS=appctlr#1[,appctlr#2,...]              ] \
[ SHARED_TAPE_RESOURCES=<TAPE>[,<TAPE#2>,...]         ] \
[ VOLUME_GROUP=<VG>[,<VG#2>,...]                      ] \
[ FORCED_VARYON={true|false}                          ] \
[ VG_AUTO_IMPORT={true|false}                         ] \
[ FILESYSTEM=/file_system#1[,/file_system#2,...]      ] \
[ DISK=<raw_disk>[,<raw_disk#2>,...]                  ] \
[ FS_BEFORE_IPADDR={true|false}                       ] \
[ WPAR_NAME="wpar_name"                               ] \
[ EXPORT_FILESYSTEM=/expfs#1[,/expfs#2,...]           ] \
[ EXPORT_FILESYSTEM_V4=/expfs#1[,/expfs#2,...]        ] \
[ STABLE_STORAGE_PATH="/fs3"                          ] \
[ NFS_NETWORK="nfs_network"                           ] \
[ MOUNT_FILESYSTEM=/nfs_fs1;/expfs1,/nfs_fs2;,...     ] \
[ MIRROR_GROUP=<replicated_resource>                  ] \
[ FALLBACK_AT=<FALLBACK_TIMER>                        ]

STARTUP:
    OHN ----- Online Home Node (default value)
    OFAN ---- Online on First Available Node
    OAAN ---- Online on All Available Nodes (concurrent)
    OUDP ---- Online Using Node Distribution Policy

FALLOVER:
    FNPN ---- Fallover to Next Priority Node (default value)
    FUDNP --- Fallover Using Dynamic Node Priority
    BO ------ Bring Offline (On Error Node Only)

FALLBACK:
    NFB ----- Never Fallback
    FBHPN --- Fallback to Higher Priority Node (default value)

NODE_PRIORITY_POLICY:
    default - next node in the NODES list
    mem ----- node with most available memory
    disk ---- node with least disk activity
    cpu ----- node with most available CPU cycles
    least --- node where the dynamic node priority script
              returns the lowest value
    most ---- node where the dynamic node priority script
              returns the highest value
```

Note: The NODE_PRIORITY_POLICY policy can only be established if the FALLOVER policy has been set to FUDNP.

```
SITE_POLICY:
     ignore -- Ignore
     primary - Prefer Primary Site
     either -- Online On Either Site
     both ---- Online On Both Sites
```

```
clmgr modify resource_group <resource_group>              \
      [ NAME=<new_resource_group_label>                ] \
      [ NODES=nodeA1[,nodeA2,...]                      ] \
      [ SECONDARYNODES=nodeB2[,nodeB1,...]             ] \
      [ SITE_POLICY={ignore|primary|either|both}       ] \
      [ STARTUP={OHN|OFAN|OAAN|OUDP}                   ] \
      [ FALLOVER={FNPN|FUDNP|BO}                       ] \
      [ FALLBACK={NFB|FBHPN}                           ] \
      [ FALLBACK_AT=<FALLBACK_TIMER>                   ] \
      [ NODE_PRIORITY_POLICY={default|mem|cpu|           \
                              disk|least|most}         ] \
      [ NODE_PRIORITY_POLICY_SCRIPT=</path/to/script>  ] \
      [ NODE_PRIORITY_POLICY_TIMEOUT=###               ] \
      [ SERVICE_LABEL=service_ip#1[,service_ip#2,...]  ] \
      [ APPLICATIONS=appctlr#1[,appctlr#2,...]         ] \
      [ VOLUME_GROUP=volume_group#1[,volume_group#2,...]] \
      [ FORCED_VARYON={true|false}                     ] \
      [ VG_AUTO_IMPORT={true|false}                    ] \
      [ FILESYSTEM=/file_system#1[,/file_system#2,...] ] \
      [ DISK=<raw_disk>[,<raw_disk#2>,...]             ] \
      [ FS_BEFORE_IPADDR={true|false}                  ] \
      [ WPAR_NAME="wpar_name"                          ] \
      [ EXPORT_FILESYSTEM=/expfs#1[,/expfs#2,...]      ] \
      [ EXPORT_FILESYSTEM_V4=/expfs#1[,/expfs#2,...]   ] \
      [ STABLE_STORAGE_PATH="/fs3"                     ] \
      [ NFS_NETWORK="nfs_network"                      ] \
      [ MOUNT_FILESYSTEM=/nfs_fs1;/expfs1,/nfs_fs2;,... ] \
      [ MIRROR_GROUP=<replicated_resource>             ] \
      [ FALLBACK_AT=<FALLBACK_TIMER>                   ]
```

Note: The appctlr value is an abbreviation for application_controller.

```
clmgr query resource_group [ <resource_group>[,<rg#2>,...] ]
clmgr delete resource_group {<resource_group>[,<rg#2>,...] |
                             ALL}
clmgr online { resource_group <resource_group>[,<rg#2>,...] | ALL} \
      [ NODES={<node>[,<node#2>,...] | ALL}]
clmgr offline resource_group {<resource_group>[,<rg#2>,...] | ALL} \
      [ NODES={<node>[,<node#2>,...] | ALL} ]
```

Note: The special ALL target for the NODES attribute is only applicable to concurrent resource groups.

```
clmgr move resource_group <resource_group>[,<rg#2>,...] \
      {NODE|SITE}=<node_or_site_label> \
      [ SECONDARY={false|true} ] \
      [ STATE={online|offline} ] \
```

Note: The *SITE* and *SECONDARY* attributes are only applicable when sites are configured in the cluster. The resource group *STATE* remains unchanged if *STATE* is not explicitly specified. An alias for *resource_group* is rg.

Fallback timer

```
clmgr add fallback_timer <timer> \
      [ YEAR=<####> ] \
      [ MONTH=<{1..12 | Jan..Dec}> ] \
      [ DAY_OF_MONTH=<{1..31}> ] \
      [ DAY_OF_WEEK=<{0..6 | Sun..Sat}> ] \
        HOUR=<{0..23}> \
        MINUTE=<{0..59}>
clmgr modify fallback_timer <timer> \
      [ YEAR=<{####}> ] \
      [ MONTH=<{1..12 | Jan..Dec}> ] \
      [ DAY_OF_MONTH=<{1..31}> ] \
```

```
      [ DAY_OF_WEEK=<{0..6 | Sun..Sat}> ] \
      [ HOUR=<{0..23}> ] \
      [ MINUTE=<{0..59}> ] \
      [ REPEATS=<{0,1,2,3,4 |
        Never,Daily,Weekly,Monthly,Yearly}> ]
clmgr query fallback_timer [<timer>[,<timer#2>,...] ]
clmgr delete fallback_timer {<timer>[,<timer#2>,...] |\
                            ALL}
```

Note: Aliases for *fallback_timer* are fa and timer.

Persistent IP/Label

```
clmgr add persistent_ip <persistent_IP> \
      NETWORK=<network> \
      [ {NETMASK=< 255.255.255.0 | PREFIX=1..128} ] \ ]
      [ NODE=<node>   ]
clmgr modify persistent_ip <persistent_label> \
      [ NAME=<new_persistent_label> ] \
      [ NETWORK=<new_network> ] \
      [ NETMASK=<node> 255.255.255.0 | PREFIX=1..128} ] \ ]
```

Note: Any value provided for NETMASK or PREFIX is ignored unless the underlying network uses a different protocol (IPv4 versus IPv6). In that case, the NETMASK or PREFIX is required.

```
clmgr query persistent_ip [ <persistent_IP>[,<pIP#2>,...] ]
clmgr delete persistent_ip {<persistent_IP>[,<pIP#2>,...] |
                            ALL}
clmgr move persistent_ip <persistent_IP> \
                         INTERFACE=<new_interface>
```

Note: An alias for *persistent_ip* is pe.

Service IP/Label

```
clmgr add service_ip <service_ip> \
      NETWORK=<network> \
      [ {NETMASK=<255.255.255.0 | PREFIX=1..128} ] \
      [ HWADDR=<new_hardware_address> ] \
      [ SITE=<new_site> ]
clmgr modify service_ip <service_ip> \
      [ NAME=<new_service_ip> ] \
      [ NETWORK=<new_network> ] \
      [ {NETMASK=<###.###.###.###> | PREFIX=1..128} ] \
      [ HWADDR=<new_hardware_address> ] \
      [ SITE=<new_site> ]
clmgr query service_ip [ <service_ip>[,<service_ip#2>,...] ]
clmgr delete service_ip {<service_ip>[,<service_ip#2>,,...] | ALL}
clmgr move service_ip <service_ip> \
      INTERFACE=<new_interface>
```

Note: If the NETMASK/PREFIX attributes are not specified, the netmask or prefix value for the underlying network is used. An alias for *service_ip* is si.

Application controller

```
clmgr add application_controller <application_controller> \
      STARTSCRIPT="/path/to/start/script" \
      STOPSCRIPT ="/path/to/stop/script" \
      [ MONITORS=<monitor>[,<monitor#2>,...] ] \
      [ STARTUP_MODE={background|foreground}
clmgr modify application_controller <application_controller> \
      [ NAME=<new_application_controller_label> ] \
      [ STARTSCRIPT="/path/to/start/script" ] \
```

```
      [ STOPSCRIPT ="/path/to/stop/script" ] \
      [ MONITORS=<monitor>[,<monitor#2>,...] ] \
      [ STARTUP_MODE={background|foreground}
clmgr query application_controller [ <appctlr>[,<appctlr#2>,...] ]
clmgr delete application_controller {<appctlr>[,<appctlr#2>,...] | \
                                     ALL}
clmgr manage application_controller {suspend|resume} \
      <application_controller> \
      RESOURCE_GROUP=<resource_group>
clmgr manage application_controller {suspend|resume} ALL
```

Note: The *appctlr* value is an abbreviation for *application_controller*. Aliases for *application_controller* are ac and app.

Application monitor

```
clmgr add application_monitor <monitor> \
      TYPE=Process \
      MODE={longrunning|startup|both} \
      PROCESSES="pmon1,dbmon,..."  \
      OWNER="<processes_owner_name>" \
      [ APPLICATIONS=<appctlr#1>[,<appctlr#2>,...] ] \
      [ STABILIZATION="1 .. 3600" ] \
      [ RESTARTCOUNT="0 .. 100" ] \
      [ FAILUREACTION={notify|fallover} ] \
      [ INSTANCECOUNT="1 .. 1024" ] \
      [ RESTARTINTERVAL="1 .. 3600" ] \
      [ NOTIFYMETHOD="</script/to/notify>" ] \
      [ CLEANUPMETHOD="</script/to/cleanup>" ] \
      [ RESTARTMETHOD="</script/to/restart>" ]

clmgr add application_monitor <monitor> \
      TYPE=Custom \
      MODE={longrunning|startup|both} \
      MONITORMETHOD="/script/to/monitor" \
      [ APPLICATIONS=<appctlr#1>[,<appctlr#2>,...] ] \
      [ STABILIZATION="1 .. 3600" ] \
      [ RESTARTCOUNT="0 .. 100" ] \
      [ FAILUREACTION={notify|fallover} ] \
      [ MONITORINTERVAL="1 .. 1024" ] \
      [ HUNGSIGNAL="1 .. 63" ] \
      [ RESTARTINTERVAL="1 .. 3600"  ] \
      [ NOTIFYMETHOD="</script/to/notify>" ] \
      [ CLEANUPMETHOD="</script/to/cleanup>" ] \
      [ RESTARTMETHOD="</script/to/restart>" ]
```

Note: STABILIZATION defaults to 180. RESTARTCOUNT defaults to 3

```
clmgr modify application_monitor <monitor> \
      [ See the "add" action, above, for a list
        of supported modification attributes. ]
clmgr query application_monitor [ <monitor>[,<monitor#2>,...] ]
clmgr delete application_monitor {<monitor>[,<monitor#2>,...] | ALL}
```

Note: The *appctlr* value is an abbreviation for *application_controller*. Aliases for *application_monitor* are am and mon.

Dependency

```
# Temporal Dependency (parent ==> child)
clmgr add dependency \
      PARENT=<rg#1> \
      CHILD="<rg#2>[,<rg#2>,...]"
clmgr modify dependency <parent_child_dependency> \
      [ TYPE=PARENT_CHILD ] \
      [ PARENT=<rg#1> ] \
```

```
      [ CHILD="<rg#2>[,<rg#2>,...]" ]

# Temporal Dependency (start/stop after)
clmgr add dependency \
      {STOP|START}="<rg#2>[,<rg#2>,...]" \
      AFTER=<rg#1>
clmgr modify dependency \
      [ TYPE={STOP_AFTER|START_AFTER} ] \
      [ {STOP|START}="<rg#2>[,<rg#2>,...]" ] \
      [ AFTER=<rg#1> ]

# Location Dependency (colocation)
clmgr add dependency \
      SAME={NODE|SITE } \
      GROUPS="<rg1>,<rg2>[,<rg#n>,...]"
clmgr modify dependency <colocation_dependency> \
      [ TYPE={SAME_NODE|SAME_SITE} ] \
      GROUPS="<rg1>,<rg2>[,<rg#n>,...]"

# Location Dependency (anti-colocation)
clmgr add dependency \
      HIGH="<rg1>,<rg2>,..." \
      INTERMEDIATE="<rg3>,<rg4>,..." \
      LOW="<rg5>,<rg6>,..."
clmgr modify dependency <anti-colocation_dependency> \
      [ TYPE=DIFFERENT_NODES ] \
      [ HIGH="<rg1>,<rg2>,..." ] \
      [ INTERMEDIATE="<rg3>,<rg4>,..." ] \
      [ LOW="<rg5>,<rg6>,..." ]

# Acquisition/Release Order
clmgr add dependency \
      TYPE={ACQUIRE|RELEASE} \
      { SERIAL="{<rg1>,<rg2>,...|ALL}" |
      PARALLEL="{<rg1>,<rg2>,...|ALL}" }
clmgr modify dependency \
      TYPE={ACQUIRE|RELEASE} \
      { SERIAL="{<rg1>,<rg2>,...|ALL}" |
      PARALLEL="{<rg1>,<rg2>,...|ALL}" }

clmgr query dependency [ <dependency> ]
clmgr delete dependency {<dependency> | ALL} \
      [ TYPE={PARENT_CHILD|STOP_AFTER|START_AFTER| \
      SAME_NODE|SAME_SITE|DIFFERENT_NODES} ]
clmgr delete dependency RESOURCE_GROUP=<RESOURCE_GROUP>
```

Note: An alias for *dependency* is de.

Tape

```
clmgr add tape <tape> \
      DEVICE=<tape_device_name> \
      [ DESCRIPTION=<tape_device_description> ] \
      [ STARTSCRIPT="</script/to/start/tape/device>" ] \
      [ START_SYNCHRONOUSLY={no|yes} ] \
      [ STOPSCRIPT="</script/to/stop/tape/device>" ] \
      [ STOP_SYNCHRONOUSLY={no|yes} ]
clmgr modify tape <tape> \
      [ NAME=<new_tape_label> ] \
      [ DEVICE=<tape_device_name> ] \
      [ DESCRIPTION=<tape_device_description> ] \
      [ STARTSCRIPT="</script/to/start/tape/device>" ] \
      [ START_SYNCHRONOUSLY={no|yes} ] \
      [ STOPSCRIPT="</script/to/stop/tape/device>" ] \
      [ STOP_SYNCHRONOUSLY={no|yes} ]
clmgr query tape [ <tape>[,<tape#2>,...] ]
clmgr delete tape {<tape> | ALL}
```

Note: An alias for *tape* is tp.

File collection

```
clmgr add file_collection <file_collection> \
      FILES="/path/to/file1,/path/to/file2,..." \
      [ SYNC_WITH_CLUSTER={no|yes}  ] \
      [ SYNC_WHEN_CHANGED={no|yes} ] \
      [ DESCRIPTION="<file_collection_description>" ]
clmgr modify file_collection <file_collection> \
      [ NAME="<new_file_collection_label>" ] \
      [ ADD="/path/to/file1,/path/to/file2,..." ] \
      [ DELETE={"/path/to/file1,/path/to/file2,..."|ALL} ] \
      [ REPLACE={"/path/to/file1,/path/to/file2,..."|""} ] \
      [ SYNC_WITH_CLUSTER={no|yes}  ] \
      [ SYNC_WHEN_CHANGED={no|yes} ] \
      [ DESCRIPTION="<file_collection_description>" ]
clmgr query file_collection [ <file_collection>[,<fc#2>,...]]
clmgr delete file_collection {<file_collection>[,<fc#2>,...]|
                              ALL}
clmgr sync file_collection <file_collection>
```

Note: The REPLACE attribute replaces all existing files with the specified set. Aliases for *file_collection* are fc and fi.

Snapshot

```
clmgr add snapshot <snapshot> \
      DESCRIPTION="<snapshot_description>" \
      [ METHODS="method1,method2,..." ]
clmgr add snapshot <snapshot> TYPE="xml"
clmgr modify snapshot <snapshot> \
      [ NAME="<new_snapshot_label>" ] \
      [ DESCRIPTION="<snapshot_description>" ]
clmgr query snapshot [ <snapshot>[,<snapshot#2>,...] ]
clmgr view snapshot <snapshot> \
      [ TAIL=<number_of_trailing_lines> ] \
      [ HEAD=<number_of_leading_lines> ] \
      [ FILTER=<pattern>[,<pattern#2>,...] ] \
      [ DELIMITER=<alternate_pattern_delimiter> ] \
      [ CASE={insensitive|no|off|false} ]
clmgr delete snapshot {<snapshot>[,<snapshot#2>,...] |
                       ALL}
clmgr manage snapshot restore <snapshot> \
      [ CONFIGURE={yes|no} ] \
      [ FORCE={no|yes} ]
```

Note: The view action displays the contents of the .info file for the snapshot, if that file exists. Aliases for *snapshot* are sn and ss.

```
clmgr manage snapshot restore <snapshot> \
      NODES=<HOST>,<HOST#2> \
      REPOSITORIES=<DISK>[,<BACKUP>][:<DISK>[,<BACKUP>]] \
      [ CLUSTER_NAME=<NEW_CLUSTER_LABEL> ] \
      [ CONFIGURE={yes|no} ] \
      [ FORCE={no|yes} ]
```

Note: For the REPOSITORIES option, any disks specified after the colon are applied to the second site. When you restore a linked cluster snapshot, any disks specified after the colon in the REPOSITORIES option are applied to the second site.

Method

```
clmgr add method <method_label> \
      TYPE=snapshot \
      FILE=<executable_file> \
```

```
      [ DESCRIPTION=<description> ]
clmgr add method <method_label> \
      TYPE=verify \
      FILE=<executable_file> \
      [ SOURCE={script|library} ] \
      [ DESCRIPTION=<description> ]
clmgr modify method <method_label> \
      TYPE={snapshot|verify} \
      [ NAME=<new_method_label> ] \
      [ DESCRIPTION=<new_description> ] \
      [ FILE=<new_executable_file> ]
clmgr add method <method_label> \
      TYPE=notify \
      CONTACT=<number_to_dial_or_email_address> \
      EVENT=<event>[,<event#2>,...] \
      [ NODES=<node>[,<node#2>,...] ] \
      [ FILE=<message_file> ] \
      [ DESCRIPTION=<description> ] \
      [ RETRY=<retry_count> ] \
      [ TIMEOUT=<timeout> ]
```

Note: NODES defaults to the local node.

```
clmgr modify method <method_label> \
      TYPE=notify \
      [ NAME=<new_method_label> ] \
      [ DESCRIPTION=<description> ] \
      [ FILE=<message_file> ] \
      [ CONTACT=<number_to_dial_or_email_address> ] \
      [ EVENT=<cluster_event_label> ] \
      [ NODES=<node>[,<node#2>,...] ] \
      [ RETRY=<retry_count> ] \
      [ TIMEOUT=<timeout> ]

clmgr query method [ <method>[,<method#2>,...] ] \
      [ TYPE={notify|snapshot|verify} ]
clmgr delete method {<method>[,<method#2>,...] | ALL} \
      [ TYPE={notify|snapshot|verify} ]
clmgr verify method <method>
```

Note: The verify action can only be applied to notify methods. If more than one method exploits the same event, and that event is specified, then both methods will be invoked. An alias for *method* is me.

Log

```
clmgr modify logs ALL DIRECTORY="<new_logs_directory>"
clmgr modify log {<log>|ALL} \
      [ DIRECTORY="{<new_log_directory>"|DEFAULT} ]
      [ FORMATTING={none|standard|low|high} ] \
      [ TRACE_LEVEL={low|high} ]
      [ REMOTE_FS={true|false} ]
clmgr query log [ <log>[,<log#2>,...] ]
clmgr view log [ {<log>|EVENTS} ] \
      [ TAIL=<number_of_trailing_lines> ] \
      [ HEAD=<number_of_leading_lines> ] \
      [ FILTER=<pattern>[,<pattern#2>,...] ] \
      [ DELIMITER=<alternate_pattern_delimiter> ] \
      [ CASE={insensitive|no|off|false} ]
clmgr manage logs collect \
      [ DIRECTORY="<directory_for_collection>" ] \
      [ NODES=<node>[,<node#2>,...] ] \
      [ RSCT_LOGS={yes|no} ] \
```

Note: When DEFAULT is specified for the DIRECTORY attribute, then the original, default PowerHA SystemMirror directory value is restored

The FORMATTING attribute only applies to the hacmp.out log, and is ignored for all other logs. The FORMATTING and TRACE_LEVEL attributes only apply to the hacmp.out and clstrmgr.debug logs, and are ignored for all other logs.

When ALL is specified in place of a log name, then the provided DIRECTORY and REMOTE_FS modifications are applied to all the logs

When EVENTS is specified in place of a log name, then an events summary report is displayed.

Volume group

```
clmgr add volume_group [ <vgname> ] \
         NODES="<node#1>,<node#2>[,...>]" \
         PHYSICAL_VOLUMES="<disk#1>[,<disk#2>,...]" \
         [ TYPE={original|big|scalable|legacy} ] \
         [ RESOURCE_GROUP=<RESOURCE_GROUP> ] \
         [ PPART_SIZE={4|1|2|8|16|32|64|128|256|512|1024} ] \
         [ MAJOR_NUMBER=## ] \
         [ ACTIVATE_ON_RESTART={false|true} ] \
         [ QUORUM_NEEDED={true|false} ] \
         [ LTG_SIZE=### ] \
         [ MIGRATE_FAILED_DISKS={false|one|pool|remove} ] \
         [ MAX_PHYSICAL_PARTITIONS={32|64|128|256|
                                    512|768|1024} ] \
         [ MAX_LOGICAL_VOLUMES={256|512|1024|2048} ] \
         [ STRICT_MIRROR_POOLS={no|yes|super} ] \
         [ MIRROR_POOL_NAME="<mp_name>" ] \
         [ CRITICAL={false|true} ] \
             [ FAILUREACTION={halt|notify|fence|
                              stoprg|moverg} ] \
             [ NOTIFYMETHOD=</file/to/invoke> ]
```

Note: Setting the volume group major number might result in the command being unable to execute successfully on a node that does not have the major number currently available. Check for a commonly available major number on all nodes before changing this setting.

```
clmgr modify volume_group <vgname> \
    [ ADD=<disk>,<disk#n>,... [ MIRROR_POOL_NAME="<mp_name>" ] ] \
    [ REMOVE=<disk>,<disk#n>,... ] \
    [ TYPE={big|scalable}] \
    [ ACTIVATE_ON_RESTART={false|true} ] \
    [ QUORUM_NEEDED={true|false} ] \
    [ LTG_SIZE=### ] \
    [ MIGRATE_FAILED_DISKS={false|one|pool|remove} ] \
    [ MAX_PHYSICAL_PARTITIONS={32|64|128|256| 512|768|1024} ] \
    [ MAX_LOGICAL_VOLUMES={256|512|1024|2048} ] \
    [ STRICT_MIRROR_POOLS={off|on|super} ] \
    [ CRITICAL={false|true}] \
    [ FAILUREACTION={halt|notify|fence| stoprg|moverg} ] \
    [ NOTIFYMETHOD="</file/to/invoke>" ] \
    [ SCSIPR_ACTION=<clear> ]
```

Note: If ENHANCED_CONCURRENT_MODE is set to false, fast disk takeover is automatically established.

MAX_PHYSICAL_PARTITIONS, MAX_LOGICAL_VOLUMES, and MIRROR_POOL_NAME only apply to scalable volume groups.

```
clmgr query volume_group [ <vg#1>[,<vg#2>,...] ]
clmgr delete volume_group
      {<volume_group> [,<vg#2>,...] | ALL }
clmgr discover volume_groups
```

Note: An alias for *volume_group* is vg.

Logical volume

```
clmgr add logical_volume [ <lvname> ] \
      VOLUME_GROUP=<vgname> \
      LOGICAL_PARTITIONS=## \
      [ DISKS="<disk#1>[,<disk#2>,...]" ] \
      [ TYPE={jfs|jfs2|sysdump|paging|
        jfslog|jfs2log|aio_cache|boot} ] \
      [ POSITION={outer_middle|outer_edge|center|
        inner_middle|inner_edge  } ] \
      [ PV_RANGE={minimum|maximum} ] \
      [ MAX_PVS_FOR_NEW_ALLOC=## ] \
      [ LPART_COPIES={1|2|3} ] \
      [ WRITE_CONSISTENCY={active|passive|off} ] \
      [ LPARTS_ON_SEPARATE_PVS={yes|no|superstrict} ] \
      [ RELOCATE={yes|no} ] \
      [ LABEL="<label>" ] \
      [ MAX_LPARTS=#### ] \
      [ BAD_BLOCK_RELOCATION={yes|no} ] \
      [ SCHEDULING_POLICY={parallel|sequential
        |parallel_sequential
        |parallel_round_robin} ] \
      [ VERIFY_WRITES={false|true} ] \
      [ ALLOCATION_MAP=<file> ] \
      [ STRIPE_SIZE={4K|8K|16K|32K|64K|128K|256K|512K|
        1M|2M|4M|8M|16M|32M|64M|128M} ] \
      [ SERIALIZE_IO={false|true} ] \
      [ FIRST_BLOCK_AVAILABLE={false|true} ] \
      [ FIRST_COPY_MIRROR_POOL=<mirror_pool> ] \
      [ SECOND_COPY_MIRROR_POOL=<mirror_pool> ] \
      [ THIRD_COPY_MIRROR_POOL=<mirror_pool> ] \
      [ GROUP=<group> ] \
      [ PERMISSIONS=<####> ] \
      [ NODE=<reference_node_in_vg> ]
```

Note: STRIPE_SIZE may not be used with LPARTS_ON_SEPARATE_PVS, PV_RANGE, or SCHEDULING_POLICY.

```
clmgr query logical_volume [ <lvname>[,<LV#2>,...] ]
clmgr delete logical_volume {[ <lv#1>[,<LV#2>,...]  ] | ALL}
```

Note: An alias for *logical_volume* is lv.

File system

```
clmgr add file_system <fsname> \
      VOLUME_GROUP=<group> \
      TYPE=enhanced \
      UNITS=### \
        [ SIZE_PER_UNIT={megabytes|gigabytes|512bytes} ] \
      [ PERMISSIONS={rw|ro} ]
      [ OPTIONS={nodev,nosuid,all} ] \
      [ BLOCK_SIZE={4096|512|1024|2048} ] \
      [ LV_FOR_LOG={ <lvname> | "INLINE" } ] \
        [ INLINE_LOG_SIZE=#### ] \
      [ EXT_ATTR_FORMAT={v1|v2} ] \
      [ ENABLE_QUOTA_MGMT={no|all|user|group} ] \
      [ ENABLE_EFS={false|true} ]
```

Note:

1. *BLOCK_SIZE* is in bytes. *LOG_SIZE* is in megabytes.
2. *LOG_SIZE* and *LV_FOR_LOG* can only be used if *INLINE_LOG* is set to true.
3. The size for an enhanced file system is 16 MB.

```
clmgr add file_system <fsname> \
      TYPE=enhanced \
      LOGICAL_VOLUME=<logical_volume> \
      [ PERMISSIONS={rw|ro} ] \
      [ OPTIONS={nodev,nosuid,all} ] \
      [ BLOCK_SIZE={4096|512|1024|2048} ] \
      [ LV_FOR_LOG={ <lvname> | "INLINE" } ] \
      [ INLINE_LOG_SIZE=#### ] \
      [ EXT_ATTR_FORMAT={v1|v2} ] \
      [ ENABLE_QUOTA_MGMT={no|all|user|group} ] \
      [ ENABLE_EFS={false|true} ]
clmgr add file_system <fsname> \
      VOLUME_GROUP=<group> \
      TYPE={standard|compressed|large} \
      UNITS=### \
         [ SIZE_PER_UNIT={megabytes|gigabytes|512bytes} ] \
      [ PERMISSIONS={rw|ro} ] \
      [ OPTIONS={nodev|nosuid|all} ] \
      [ DISK_ACCOUNTING={false|true} ] \
      [ FRAGMENT_SIZE={4096|512|1024|2048} ] \
      [ BYTES_PER_INODE={4096|512|1024|2048|8192|
                         16384|32768|65536|131072} ] \
      [ ALLOC_GROUP_SIZE={8|16|32|64} ] \
      [ LV_FOR_LOG=<lvname> ]
```

Note: FRAGMENT_SIZE is only valid for standard and compressed file systems.

```
clmgr add file_system <fsname> \
      TYPE={standard|compressed|large} \
      LOGICAL_VOLUME=<logical_volume> \
      [ PERMISSIONS={rw|ro} ] \
      [ OPTIONS={nodev|nosuid|all} ] \
      [ DISK_ACCOUNTING={false|true} ] \
      [ FRAGMENT_SIZE={4096|512|1024|2048} ] \
      [ BYTES_PER_INODE={4096|512|1024|2048|8192|
                         16384|32768|65536|131072} ] \
      [ ALLOC_GROUP_SIZE={8|16|32|64} ] \
      [ LV_FOR_LOG=<lvname> ]

clmgr query file_system [ <fs#1>[,<fs#2>,...] ]
clmgr delete file_system { <fsname>[,<FS#2>,...] | ALL } \
      [ REMOVE_MOUNT_POINT={false|true} ]
```

Note: An alias for *file_system* is fs.

Physical volume

```
clmgr query physical_volume \
      [ <disk>[,<disk#2>,...] ] \
      [ NODES=<node>,<node#2>[,<node#3>,...] ] \
      [ TYPE={available|all|tiebreaker} ]
```

Note: Node can be either a node name or a network-resolvable name, for example, host name or IP address.

Disk can be either a device name (hdisk0) or a PVID (00c3a28ed9aa3512).

```
clmgr modify physical_volume <disk_name_or_PVID> \
      NAME=<new_disk_name> \
      [ NODE=<reference_node> ] \
      [ ALL_NODES={false|true} ] \
      [ SCSIPR_ACTION={clear} ]
```

Note: The NODE attribute is required if the specified disk is provided using a device name, such as hdisk#. If the disk is specified using the PVID, you do not need to reference the NODE attribute.

An alias for *physical_volume* is pv.

Mirror pool

```
clmgr add mirror_pool <pool_name> \
      VOLUME_GROUP=<vgname> \
      [ PHYSICAL_VOLUMES="<disk#1>[,<disk#2>,...]" ] \
      [ MODE={sync|async} ] \
      [ ASYNC_CACHE_LV=<lvname> ] \
      [ ASYNC_CACHE_HW_MARK=## ]

clmgr add mirror_pool <pool_name> \
      [ VOLUME_GROUP=<vgname> ] \
      PHYSICAL_VOLUMES="<disk>[,<disk#2>,...]"
```

Note: If an *add* operation is performed on an existing mirror pool, the specified physical volumes are added to that mirror pool.

```
clmgr modify mirror_pool <pool_name> \
      [ VOLUME_GROUP=<vgname> ]\
      [ NAME=<new_pool_name> ] \
      [ MODE={sync|async} ] \
      [ FORCE_SYNC={false|true} ] \
      [ ASYNC_CACHE_LV=<lvname> ] \
      [ ASYNC_CACHE_HW_MARK=## ]

clmgr query mirror_pool [ <pool_name>[,<pool#2>,...] ]
clmgr delete mirror_pool <pool_name>,[,<pool#2>,...]| ALL }\
      [ VOLUME_GROUP=<vgname> ]
clmgr delete mirror_pool <pool_name> \
      [ VOLUME_GROUP=<vgname> ] \
      PHYSICAL_VOLUMES="<disk>[,<disk#2>,...]"
```

Note: When physical volumes are specified for a delete operation, the list of disks will be removed from the mirror pool. If all disks are removed, the mirror pool is removed.

Note: An aliases for *mirror_pool* are mp and pool.

EFS

```
clmgr add efs \
      MODE=ldap \
      [ PASSWORD=<password> ]
clmgr add efs \
      MODE=shared_fs \
      VOLUME_GROUP=<vgname> \
      SERVICE_IP=<service_ip> \
      [ PASSWORD=<password> ]

clmgr modify efs \
      MODE={ldap|shared_fs} \
      [ VOLUME_GROUP=<vgname> ] \
      [ SERVICE_IP=<service_ip> ] \
      [ PASSWORD=<password> ]

clmgr query efs
clmgr delete efs
```

Report

```
clmgr view report [<report>] \
      [ FILE=<PATH_TO_NEW_FILE> ] \
      [ TYPE={text|html} ]

clmgr view report {nodeinfo|rginfo|lvinfo|
      fsinfo|vginfo|dependencies} \
      [ TARGETS=<target>[,<target#2>,...] ] \
      [ FILE=<PATH_TO_NEW_FILE> ] \
      [ TYPE={text|html} ]

clmgr view report cluster \
      TYPE=html \
      [ FILE=<PATH_TO_NEW_FILE> ] \
      [ COMPANY_NAME="<BRIEF_TITLE>" ] \
      [ COMPANY_LOGO="<RESOLVEABLE_FILE>" ]

clmgr view report availability \
      [ TARGETS=<appctlr>[,<appctlr#2>,...] ] \
      [ FILE=<PATH_TO_NEW_FILE> ] \
      [ TYPE={text|html} ] \
      [ BEGIN_TIME="YYYY:MM:DD" ] \
     [ END_TIME="YYYY:MM:DD" ]
```

Note: The currently supported reports are basic, cluster, status, topology, applications, availability, events, nodeinfo, rginfo, networks, vginfo, lvinfo, fsinfo, dependencies, and roha. Some of these reports provide overlapping information, but each also provides its own, unique information, as well.

The appctlr value is an abbreviation for application_controller.

MM must be 1 - 12. DD must be 1 - 31.

If no BEGIN_TIME is provided, then a report will be generated for the last 30 days prior to END_TIME.

If no END_TIME is provided, then the current time will be the default.

An alias for *report* is re.

LDAP server

The following syntax is used for configuring one or more LDAP servers for the cluster.

```
clmgr add ldap_server <server>[,<server#2>,...] \
      ADMIN_DN=<admin_distinguished_name> \
      PASSWORD=<admin_password> \
      BASE_DN=<suffix_distinguished_name> \
      SSL_KEY=<full_path_to_key> \
      SSL_PASSWORD=<SSL_key_password> \
      VERSION=<version> \
      DB2_INSTANCE_PASSWORD=<password> \
      ENCRYPTION_SEED=<seed> \
      [ SCHEMA=<schema_type> ] \
      [ PORT={636|###} ]
```

Note: An alias for *ldap_server* is ls.

The following syntax is used for adding one ore more LDAP servers that is already configured to the cluster.

```
clmgr add ldap_server <server>[,<server#2>,...] \
      ADMIN_DN=<admin_distinguished_name> \
      PASSWORD=<admin_password> \
```

```
      BASE_DN=<suffix_distinguished_name> \
      SSL_KEY=<full_path_to_key> \
      SSL_PASSWORD=<SSL_key_password> \
      [ PORT={636|###} ]
```

Note: If more than one server is specified, they must be in a peer-to-peer configuration, sharing the same port number.

```
clmgr query ldap_server
clmgr delete ldap_server
```

LDAP client

```
clmgr add ldap_client \
      SERVERS=<LDAP_server>[,<LDAP_server#2>]\
      BIND_DN=<bind_distinguished_name> \
        PASSWORD=<LDAP_admin_password> \
      BASE_DN=<base_dn> \
      SSL_KEY=<full_path_to_key> \
        SSL_PASSWORD=<SSL_key_password> \
      [ PORT={636|###} ] \

clmgr query ldap_client
clmgr delete ldap_client
```

Note: An alias for *ldap_client* is lc.

User

```
clmgr add/modify user <user_name> \
      [ REGISTRY={local|ldap} ] \
      [ RESOURCE_GROUP=<resource_group> ] \
      [ ID=### ] \
      [ PRIMARY=<group> ] \
      [ PASSWORD="{<password>|}" ] \
      [ CHANGE_ON_NEXT_LOGIN={true|false} ] \
      [ GROUPS=<group#1>[,<group#2>,...] ] \
      [ ADMIN_GROUPS=<group#1>[,<group#2>,...] ] \
      [ ROLES=<role#1>[,<role#2>,...] ] \
      [ SWITCH_USER={true|false} ] \
      [ SU_GROUPS={ALL|<group#1>[,<group#2>,...]} ] \
      [ HOME=<full_directory_path> ] \
      [ SHELL=<defined_in_/etc/shells> ] \
      [ INFO=<user_information> ] \
      [ EXPIRATION=<MMDDhhmmyy> ] \
      [ LOCKED={false|true} ] \
      [ LOGIN={true|false} ] \
      [ REMOTE_LOGIN={true|false} ] \
      [ SCHEDULE=<range#1>[,<range#2>,...>] ] \
      [ MAX_FAILED_LOGINS={#|0} ] \
      [ AUTHENTICATION={compat|files|DCE|ldap} ] \
      [ ALLOWED_TTYS=<tty#1>[,<tty#2>,...] ] \
      [ DAYS_TO_WARN={#|0} ] \
      [ PASSWORD_VALIDATION_METHODS=<meth#1>[,<meth#2>,...]]\
      [ PASSWORD_FILTERS=<filter#1>[,<filter#2>,...] ] \
      [ MIN_PASSWORDS=<number_of_passwords_before_reuse> ] \
      [ REUSE_TIME=<weeks_before_password_reuse> ] \
      [ LOCKOUT_DELAY=<weeks_btwn_expiration_and_lockout> ] \
      [ MAX_PASSWORD_AGE={0..52} ] \
      [ MIN_PASSWORD_LENGTH={0..8} ] \
      [ MIN_PASSWORD_ALPHAS={0..8} ] \
      [ MIN_PASSWORD_OTHERS={0..8} ] \
      [ MAX_PASSWORD_REPEATED_CHARS={0..52} ] \
      [ MIN_PASSWORD_DIFFERENT={0..8} ] \
      [ UMASK=#### ] \
```

```
      [ AUDIT_CLASSES=<class#1>[,<class#2>,...] ] \
      [ TRUSTED_PATH={nosak|on|notsh|always} ] \
      [ PRIMARY_AUTH={SYSTEM|.} ] \
      [ SECONDARY_AUTH={NONE|SYSTEM|<token>;<user>} ] \
      [ PROJECTS=<project#1>[,<project#2>,...] ] \
      [ KEYSTORE_ACCESS={file|none} ] \
          [ ADMIN_KEYSTORE_ACCESS={file|none} ] \
          [ KEYSTORE_MODE={admin|guard} ] \
          [ ALLOW_MODE_CHANGE={false|true} ] \
      [ KEYSTORE_ENCRYPTION={RSA_1024|RSA_2048|RSA_4096} ] \
          [ FILE_ENCRYPTION={AES_128_CBC|AES_128_EBC| \
                             AES_192_CBC|AES_192_ECB| \
                             AES_256_CBC|AES_256_ECB} ] \
      [ ALLOW_PASSWORD_CHANGE={no|yes} ]
```

Note: The INFO field only accepts alphanumeric characters including a space, an underscore (_), and a hyphen (-).

Note: For an *add* operation, *REGISTRY* indicates where to create the user. For *modify*, it indicates which instance of the specified user to change.

Note: SCHEDULE defines the times when the user is allowed to login to this system. The SCHEDULE value is a comma separated list of items as follows:

```
* [!][MMdd[-MMdd]]:hhmm-hhmm
* [!]MMdd[-MMdd][:hhmm-hhmm]
* [!][w[-w]]:hhmm-hhmm
* [!]w[-w][:hhmm-hhmm]
```

Where *MM* is a month number (00=January, 11=December), *dd* is the day of the month, *hh* is the hour of the day (00 - 23), *mm* is the minute of the hour, and *w* is the day of the week (0=Sunday, 6=Saturday). An exclamation point can be used to indicate that the access during the specified time range is disallowed.

MAX_FAILED_LOGINS, DAYS_TO_WARN, MIN_PASSWORDS, REUSE_TIME can be set to zero to disable these features.

LOCKOUT_DELAY can be set to -1 to disable these features.

```
clmgr modify user {<user_name> | ALL_USERS} \
      ALLOW_PASSWORD_CHANGE={no|yes}
```

Note: *ALLOW_PASSWORD_CHANGE* indicates if the user is allowed to change their password across the entire cluster using C-SPOC.

```
clmgr query user TYPE={AVAILABLE|ALLOWED}
clmgr query user RESOURCE_GROUP=<resource_group>
clmgr query user <user_name> \
      [ RESOURCE_GROUP=<resource_group> ]
clmgr delete user <user_name> \
      [ RESOURCE_GROUP=<resource_group> ] \
      [ REMOVE_AUTH_INFO={true|false} ]
      [ REGISTRY={files |LDAP} ]
```

Group

```
clmgr add group <group_name>
      [ REGISTRY={local(files)|LDAP} ]
      [ RESOURCE_GROUP=<resource_group> ] \
      [ ID=### ] \
      [ ADMINISTRATIVE={false|true} ] \
```

```
      [ USERS=<user#1>[,<user#2>,...] ] \
      [ ADMINS=<admin#1>[,<admin#2>,...] ] \
      [ PROJECTS=<project#1>[,<project#2>,...] ] \
      [ KEYSTORE_MODE={admin|guard} ] \
      [ KEYSTORE_ENCRYPTION={ RSA_1024|RSA_2048|RSA_4096} ] \
      [ KEYSTORE_ACCESS={file|none} ] \

clmgr modify group <group_name> \
      [ RESOURCE_GROUP=<resource_group> ] \
      [ ID=### ] \
      [ ADMINISTRATIVE={false|true} ] \
      [ USERS=<user#1>[,<user#2>,...] ] \
      [ ADMINS=<admin#1>[,<admin#2>,...] ] \
      [ PROJECTS=<project#1>[,<project#2>,...] \
      [ KEYSTORE_MODE={admin|guard} ] \
      [ KEYSTORE_ENCRYPTION={ RSA_1024|RSA_2048|RSA_4096} ] \
      [ KEYSTORE_ACCESS={file|none} ]
```

Note: The RG option is required for locally defined groups. If the RG option is not provided, it is assumed there is an LDAP group.

```
clmgr query group RESOURCE_GROUP=<resource_group>
clmgr query group <group_name> \
      [ RESOURCE_GROUP=<resource_group> ]

clmgr delete group <group_name> \
      [ RESOURCE_GROUP=<resource_group> ] \
      [ REGISTRY={files|LDAP} ]
```

Note: The RG option is required for locally defined groups. An alias for *group* is gp.

Storage agent

```
clmgr add storage_agent <agent_name> \
      TYPE={ds8k_gm|xiv_rm} \
      ADDRESSES=<IP>[<IP#2>,...] \
      [ USER=<user_id> ] \
      [ PASSWORD=<password> ] \
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]
clmgr modify storage_agent <agent_name> \
      [ NAME=<new_agent_name> ] \
      [ ADDRESSES=<IP>[<IP#2>,...] ] \
      [ USER=<user_id> ] \
      [ PASSWORD=<password> ] \
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]
clmgr query storage_agent [ <agent>[,<agent#2>,...] ]
clmgr delete storage_agent {<agent>[,<agent#2>,...] | ALL}
```

Note: An alias for *storage agent* is sta.

Storage system

```
clmgr add storage_system <storage_system_name> \
      TYPE={ds8k_gm|xiv_rm} \
      SITE=<site> \
      AGENTS=<agent>[,<agent#2>,...] \
      VENDOR_ID=<identifier> \
      [ WWNN=<world_wide_node_name> ] \
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]
clmgr add storage_system <storage_system_name> \
      TYPE=ds8k_inband_mm \
      SITE=<site> \
      VENDOR_ID=<identifier> \
      [ WWNN=<world_wide_node_name> ] \
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]
```

```
clmgr add storage_system <storage_system_name> \
      TYPE=svc \
      SITE=<site> \
      ADDRESSES=<IP>[<IP#2>,...] \
      MASTER=<Master/Auxiliary> \
      PARTNER=<Remote Partner> \
      [ AGENTS=<agent>[,<agent#2>,...] ] \
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

clmgr modify storage_system <storage_system_name> \
      [ NAME=<new_storage_system_name> ] \
      [ SITE=<site> ] \
      [ AGENTS=<agent>[,<agent#2>,...] ] \
      [ WWNN=<world_wide_node_name> ] \
      [ VENDOR_ID=<identifier> ] \
      [ ADDRESSES=<IP>[<IP#2>,...] ] \
      [ MASTER=<Master/Auxiliary> ] \
      [ PARTNER=<Remote Partner> ] \
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

clmgr query storage_system [ <storage_system>[,<ss#2>,...] ]

clmgr -a VENDOR_ID query storage_system \
      TYPE={ds8k_gm|ds8k_inband_mm|xiv_rm}
```

Note: The following query lists the available vendor IDs.

```
clmgr delete storage_system {<storage_system>[,<ss#2>,...] | ALL}
```

Note: An alias for *storage system* is sts.

Mirror pair

```
clmgr add mirror_pair <mirror_pair_name> \
      FIRST_DISK=<disk_1> \
      SECOND_DISK=<disk_2>
clmgr modify mirror_pair <mirror_pair_name> \
      [ NAME=<new_mirror_pair_name> ] \
      [ FIRST_DISK=<disk_1> ] \
      [ SECOND_DISK=<disk_2> ]
clmgr query mirror_pair [ <mirror_pair>[,<mp#2>,...] ]
clmgr delete mirror_pair {<mirror_pair>[,<mp#2>,...] | ALL}
```

Note: An alias for *mirror_pair* is mip.

Mirror group

```
: HyperSwap user mirror groups
  clmgr add mirror_group <mirror_group_name> \
        TYPE=ds8k_inband_mm \
        MG_TYPE=user \
        VOLUME_GROUPS=<volume_group>[,<vg#2>,...] \
        DISKS=<raw_disk>[,<disk#2>,...] \
        [ HYPERSWAP_ENABLED={no|yes} ] \
        [ CONSISTENT={yes|no} ] \
        [ UNPLANNED_HS_TIMEOUT=## ] \
        [ HYPERSWAP_PRIORITY={medium|high} ] \
        [ RECOVERY={manual|auto} ] \
        [ RESYNC={manual|auto} ] \
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

 clmgr modify mirror_group <mirror_group_name> \
       [ NAME=<new_mirror_group_name> ] \
       [ VOLUME_GROUPS=<volume_group>[,<vg#2>,...] ] \
       [ DISKS=<raw_disk>[,<disk#2>,...] ] \
       [ STORAGE_SYSTEMS=<storage_system>[,<ss#2>,...] ] \
       [ HYPERSWAP_ENABLED={no|yes} ] \
```

```
      [ CONSISTENT={yes|no} ] \
      [ UNPLANNED_HS_TIMEOUT=## ] \
      [ HYPERSWAP_PRIORITY={medium|high} ] \
      [ RECOVERY={manual|auto} ] \
      [ RESYNC={manual|auto} ] \
      [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

: HyperSwap system mirror groups
  clmgr add mirror_group <mirror_group_name> \
        TYPE=ds8k_inband_mm \
        MG_TYPE=system \
        VOLUME_GROUPS=<volume_group>[,<vg#2>,...] \
        DISKS=<raw_disk>[,<disk#2>,...] \
        NODE=<node> \
        HYPERSWAP_ENABLED={no|yes} \
        [ CONSISTENT={yes|no} ] \
        [ UNPLANNED_HS_TIMEOUT=## ] \
        [ HYPERSWAP_PRIORITY={medium|high} ] \
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

  clmgr modify mirror_group <mirror_group_name> \
        [ NAME=<new_mirror_group_name> ] \
        [ VOLUME_GROUPS=<volume_group>[,<vg#2>,...] ] \
        [ DISKS=<raw_disk>[,<disk#2>,...] ] \
        [ NODE=<node> ] \
        [ STORAGE_SYSTEMS=<storage_system>[,<ss#2>,...] ] \
        [ HYPERSWAP_ENABLED={no|yes} ] \
        [ CONSISTENT={yes|no} ] \
        [ UNPLANNED_HS_TIMEOUT=## ] \
        [ HYPERSWAP_PRIORITY={medium|high} ] \
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

: HyperSwap repository mirror groups
  clmgr add mirror_group <mirror_group_name> \
        TYPE=ds8k_inband_mm \
        MG_TYPE=repository \
        SITE=<site> \
        NON_HS_DISK=<Non-HyperSwap_disk> \
        HS_DISK=<HyperSwap_disk> \
        [ HYPERSWAP_ENABLED={no|yes} ] \
        [ CONSISTENT={yes|no} ] \
        [ UNPLANNED_HS_TIMEOUT=## ] \
        [ HYPERSWAP_PRIORITY={medium|high} ] \
        [ RESYNC={manual|auto} ] \
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

  clmgr modify mirror_group <mirror_group_name> \
        [ NAME=<new_mirror_group_name> ] \
        [ SITE=<node> ] \
        [ NON_HS_DISK=<non-HyperSwap_disk> ] \
        [ HS_DISK=<HyperSwap_disk> ] \
        [ STORAGE_SYSTEMS=<storage_system>[,<ss#2>,...] ] \
        [ HYPERSWAP_ENABLED={no|yes} ] \
        [ CONSISTENT={yes|no} ] \
        [ UNPLANNED_HS_TIMEOUT=## ] \
        [ HYPERSWAP_PRIORITY={medium|high} ] \
        [ RESYNC={manual|auto} ] \
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

: DS8000 Global Mirror and XIV mirror groups
  clmgr add mirror_group <mirror_group_name> \
        TYPE={ds8k_gm|xiv_rm} \
        MODE={sync|async} \
        RECOVERY={auto|manual} \
        [ STORAGE_SYSTEMS=<storage_system>[,<ss#2>,...] ] \
        [ VENDOR_ID=<vendor_specific_identifier> ] \
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]
```

```
  clmgr modify mirror_group <mirror_group_name> \
        [ NAME=<new_mirror_group_name> ] \
        [ MODE={sync|async} ] \
        [ RECOVERY={auto|manual} ] \
        [ STORAGE_SYSTEMS=<storage_system>[,<ss#2>,...] ] \
        [ VENDOR_ID=<vendor_specific_identifier> ] \
        [ ATTRIBUTES=<NAME>@<VALUE>[,<NAME#2>@<VALUE#2>,...] ]

: SVC mirror groups
  clmgr add mirror_group <mirror_group_name> \
        TYPE=svc \
        STORAGE_SYSTEMS=<MASTER_SVC>,<AUXILIARY_SVC> \
        MIRROR_PAIRS=<mirror_pair>[,<mirror_pair#2>,...] ] \
        [ MODE={sync|async} ] \
        [ RECOVERY={auto|manual} ]

  clmgr modify mirror_group <mirror_group_name> \
        [ NAME=<new_mirror_group_name> ] \
        [ STORAGE_SYSTEMS=<MASTER_SVC>,<AUXILIARY_SVC> ] \
        [ MIRROR_PAIRS=<mirror_pair>[,<mirror_pair#2>,...] ] \
        [ MODE={sync|async} ] \
        [ RECOVERY={auto|manual} ]

: Hitachi mirror groups
  clmgr add mirror_group <mirror_group_name> \
        TYPE=hitachi \
        VENDOR_ID=<device_group> \
        HORCM_INSTANCE=<instance> \
        [ MODE={sync|async} ] \
        [ RECOVERY={auto|manual} ] \
        [ HORCM_TIMEOUT=### ] \
        [ PAIR_EVENT_TIMEOUT=### ]

  clmgr modify mirror_group <mirror_group_name> \
        [ NAME=<new_mirror_group_name> ] \
        [ VENDOR_ID=<device_group> ] \
        [ HORCM_INSTANCE=<instance> ] \
        [ MODE={sync|async} ] \
        [ RECOVERY={auto|manual} ] \
        [ HORCM_TIMEOUT=### ] \
        [ PAIR_EVENT_TIMEOUT=### ]

: EMC mirror groups
  clmgr add mirror_group <mirror_group_name> \
        TYPE=emc \
        [ MG_TYPE={composite|device} ] \
        [ MODE={sync|async} ] \
        [ RECOVERY={auto|manual} ] \
        [ CONSISTENT={yes|no} ] \
        [ VENDOR_ID=<vendor_specific_identifier> ]

  clmgr modify mirror_group <mirror_group_name> \
       [ NAME=<new_mirror_group_name> ] \
       [ MG_TYPE={composite|device} ] \
       [ MODE={sync|async} ] \
       [ RECOVERY={auto|manual} ] \
       [ CONSISTENT={yes|no} ] \
       [ VENDOR_ID=<device_group> ]

: HyperSwap mirror groups
  clmgr {swap|view} mirror_group <mirror_group_name>[,<mg#2>,...] \
        [ NODE=<node_name> ]
  clmgr {swap|view} mirror_group \
        NODES=<node_name>[,<node#2>,...] \
        [ SYSTEM_GROUPS={yes|no} ]
```

```
  clmgr {swap|view} mirror_group \
        SITES=<site_name>[,<site#2>] \
        [ SYSTEM_GROUPS={yes|no} ] \
        [ REPOSITORY_GROUP={yes|no} ]
```

Note: The `swap` and `view` attributes are only valid for DS-Series Inband (HyperSwap®).

```
clmgr manage mirror_group refresh
      <mirror_group_name>[,<mg#2>,...] \
      [ NODE=<node_name> ]
clmgr manage mirror_group refresh \
      NODES=<node_name>[,<node#2>,...] \
      [ SYSTEM_GROUPS={yes|no} ]
clmgr manage mirror_group refresh \
      SITES=<site_name>[,<site#2>] \
      [ SYSTEM_GROUPS={yes|no} ] \
      [ REPOSITORY_GROUP={yes|no} ]

: All mirror groups
  clmgr query mirror_group [ <mirror_group>[,<mg#2>,...] ]
  clmgr delete mirror_group {<mirror_group>[,<mg#2>,...] | ALL}
```

Note: An alias for *mirror_group* is mig.

Event

```
cl clmgr add event <EVENT_NAME> \
         FILE=<EXECUTABLE_FILE> \
         [ DESCRIPTION=<EVENT_DESCRIPTION> ]
clmgr modify event <EVENT_NAME> \
         [ NAME=<NEW_EVENT_NAME> ] \
         [ FILE=<EXECUTABLE_FILE> ] \
         [ DESCRIPTION=<EVENT_DESCRIPTION> ]
clmgr modify event <BULTIN EVENT_NAME> \
         [ COMMAND=<COMMAND_OR_FILE> ] \
         [ NOTIFY_COMMAND=<COMMAND_OR_FILE> ] \
         [ RECOVERY_COMMAND=<COMMAND_OR_FILE> ] \
           [ RECOVERY_COUNTER=# ] \
         [ PRE_EVENT_COMMAND=<CUSTOM_EVENT> ] \
         [ POST_EVENT_COMMAND=<CUSTOM_EVENT> ]
clmgr query event [ <EVENT_NAME>[,<EVENT_NAME#2>,...] ]
         [ TYPE={CUSTOM|PREDEFINED|ALL} ]
clmgr delete event { <EVENT_NAME>[,<EVENT_NAME#2>,...] | ALL }
```

Note: An alias for *event* is ev.

HMC

```
clmgr add hmc <HMC> \
            [ TIMEOUT=<###> ] \
            [ RETRY_COUNT=<###> ] \
            [ RETRY_DELAY=<###> ] \
            [ NODES=<node>[,<node#2>,...] ] \
            [ SITES=<site>[,<site#2>,...] ] \
            [ CHECK_HMC=<Yes|No> ]
clmgr modify hmc <HMC> \
            [ TIMEOUT=<###> ] \
            [ RETRY_COUNT=<###> ] \
            [ RETRY_DELAY=<###> ] \
            [ NODES=<node>[,<node#2>,...] ] \
            [ SITES=<site>[,<site#2>,...] ] \
            [ CHECK_HMC=<Yes|No> ]

clmgr query hmc [<HMC>[,<HMC#2>,...]]
clmgr delete hmc {<HMC> | ALL}
```

Note: The **clmgr delete** example removes either the specified HMC, or all HMCs, associated with the specified node. If no nodes are specified, all nodes are removed.

CoD

```
clmgr add cod <APPCTRL> \
      [ USE_DESIRED="Yes|No"> ] \
      [ OPTIMAL_MEM=#.## ]  \
      [ OPTIMAL_CPU=# ]  \
      [ OPTIMAL_PU=#.## ] \
      [ OPTIMAL_VP=# ]

clmgr modify cod <APPCTRL> \
      [ USE_DESIRED="Yes|No"> ] \
      [ OPTIMAL_MEM=#.## ] \
      [ OPTIMAL_CPU=# ] \
      [ OPTIMAL_PU=#.## ] \
      [ OPTIMAL_VP=# ]
```

Note:

1. You can use this command to provision the optimal level of resources that are required to run the application controller.
2. If you set `USE_DESIRED=1`, the desired level of the LPAR profile that provides the optimal level of resources for the application controller is used.
3. If you set `USE_DESIRED=0`, you can be more precise and use the `OPTIMAL_MEM`, `OPTIMAL_CPU`, `OPTIMAL_PU` and `OPTIMAL_VP` values to configuring the level of resources that are required by the application controller.
4. Provisioning a level of resources for an application controller allows PowerHA SystemMirror to perform operations (DLPAR, On/Off CoD, EPCoD) that provide the optimal level of resources for the application controller.
5. You can check the level of provisioning by verifying your cluster with the **clmgr verify cluster** command.
6. Aliases for *cod* are *roha*, *dlpar*, and *cuod*.

```
clmgr query cod [<APPCTRL> ]
clmgr delete cod {<APPCTRL> | ALL}
```

Examples

In the following examples, the class attribute for the **clmgr** command is not case sensitive. For example, in the following command, the `NODES` attribute could be `NODES`, `nodes`, or `Nodes`.

```
clmgr create cluster clMain NODES=nodeA,nodeB
```

1. The following example creates a PowerHA SystemMirror Standard Edition for AIX cluster that contains two nodes named nodeA and nodeB. The cluster name is haCL, and it has a repository disk named hdisk5. The environment requires the use of a predetermined multicast address of 229.9.3.17 for the cluster.

   ```
   clmgr create cluster haCL NODES=nodeA,nodeB \
         REPOSITORY=hdisk5 \
         CLUSTER_IP=229.9.3.17
   clmgr sync cluster
   ```

   Note: The `CLUSTER_IP` attribute is required in this example only because the environment requires a multicast address. If a multicast address is not provided, the system selects an address based on the addresses currently in use at that time.
2. The following example creates a standard (nonconcurrent) resource group using default policies. The resource group is named db2RG, contains a service IP address named access1, and contains an application controller named db2Controller. The resource group manages two nonconcurrent volume groups named vg1 and vg2.

```
clmgr add resource_group db2RG SERVICE_IP=access1 \
      APPLICATIONS=db2Controller \
      VOLUME_GROUP=vg1,vg2
clmgr sync cluster
```

3. You can use the following commands to check the status of various objects inside a cluster.

   ```
   clmgr -a STATE query cluster
   clmgr -a STATE query node nodeA
   clmgr -a STATE query resource_group rg1
   ```

   Note:

   - The `STATE` class returns a logical worst-case aggregation for the entire cluster. For example, if one cluster in a four-node cluster is experiencing an error, the status returned for the entire cluster is reported as an error.
   - The value returned from running this command is in the standard `ATTR=VALUE` format. For example, if a cluster is offline, the value returned is `STATE=OFFLINE`.
   - You can retrieve multiple attributes at once by using the **-a** flag. For example, if you run the following command, you get both the name and state of the cluster:

     ```
     clmgr -a STATE,NAME query cluster
     ```

4. All actions, classes, and attributes can be shortened to either an explicitly named alias or the fewest number of characters that make them unique. The following examples display the full command on and the shortened version of the same command below it.
   - ```
     clmgr query resource_group
     clmgr q rg
     ```
   - ```
     clmgr modify node mynode PERSISTENT_IP=myIP NETWORK=myNet
     clmgr mod node mynode pe=myIP netw=myNet
     ```
   - ```
     clmgr online node nodeA
     clmgr start node nodeA
     ```

   Note: The shortening of these actions, classes, and attributes is intended for use when you are using the **clmgr** command interactively on a cluster. Although these abbreviations can be used within scripts, avoid using them inside scripts because they do not provide easily readable code.

5. Help information is provided from the command line for the **clmgr** command. If you do not know the entire command that you want to run, you can type as much as you know and help information is displayed. For example, if you provide an invalid object or value for part of the command, the help information displays only valid objects or values. Run the following commands as examples to view how different help information is displayed from the command line.

   ```
   clmgr
   clmgr view
   clmgr view report
   clmgr view report -h
   ```

   Note: You can only use the **-h** flag after either an object class or a set of option pairs that request a listing of all valid options for a particular operation. This flag is the only flag for the **clmgr** command that need not be placed immediately after the `clmgr` command.

The following examples describe some common usage scenarios of `clmgr` command. All of the examples have been tested. Substitute the value for values that are valid for your environment. The following tasks are the basis for the scenarios and are described in detail.

- Create a cluster
- Create a resource group
- Check current status
- View all attributes and settings
- Display objects based on some filter or criteria
- Make the `clmgr` command a little easier to use
   ```

- Get instant help for the clmgr command

Example: Create a standard cluster

Details:

This cluster is a standard cluster with two nodes and does not have any associated sites. The cluster name is DB2_cluster and the nodes are named DBPrimary and DBBackup. The repository disk is created on the disk named hdisk5.

Example:

```
1. clmgr create cluster DB2_cluster NODES=DBPrimary,DBBackup \
                                    REPOSITORY=hdisk5
2. clmgr sync cluster
```

Comments:

- The repository disk resolves on the node that runs the **clmgr** command. You can specify the repository disk in PVID or UUID format.
- A heartbeat type was not specified. Thus, the cluster uses the default of unicast communication.
- The **clmgr** command is not case-sensitive. You can specify the repository attribute as REPOSITORY, Repository, or repository.

Example: Create a stretched cluster

Details:

This cluster is a stretched cluster named Oracle_cluster. The cluster has four nodes named Ora1, Ora2, Ora3, and Ora4. The cluster has two sites named Ora_Primary and Ora_Secondary. The site named Ora_Primary manages the nodes named Ora1 and Ora2. The site named Ora_Secondary manages the nodes named Ora3 and Ora4. The repository disk is created on the disk named hdisk5. The cluster uses multicast communication as the heartbeat type.

Example:

```
1. clmgr create cluster Oracle_cluster \
         NODES=Ora1,Ora2,Ora3,Ora4 \
         TYPE=SC \
         REPOSITORY=hdisk5 \
         HEARTBEAT_TYPE=multicast
2. clmgr add site Ora_Primary NODES=Ora1,Ora2
3. clmgr add site Ora_Secondary NODES=Ora3,Ora4
4. clmgr sync cluster
```

Comment:

The repository disk resolves on the node that runs the **clmgr** command. You can specify the repository disk in PVID or UUID format.

Example: Create a linked cluster

Details:

This cluster is a linked cluster named SAP-cluster. The cluster has four nodes named SAP-A1, SAP-A2, SAP-B1, and SAP-B2. The cluster has two sites named SAP_Active and SAP_Backup. The site named SAP_Active manages the nodes named SAP-A1 and SAP-A2. The site named SAP_Backup manages the nodes

named SAP-B1 and SAP-B2. The repository disk on the SAP_Active site is named hdisk5. The repository disk on the SAP_Backup site is named hdisk11. The cluster uses unicast communication for the heartbeat type.

Example:

1.
```
clmgr create cluster SAP-cluster \
        NODES=SAP-A1,SAP-A2,SAP-B1,SAP-B2 \
        TYPE=LC \
        HEARTBEAT_TYPE=unicast
```
2.
```
clmgr add site SAP_Active NODES=SAP-A1,SAP-A2 REPOSITORY=hdisk5
```
3.
```
clmgr add site SAP_Backup NODES=SAP-B1,SAP-B2 REPOSITORY=hdisk11
```
4.
```
clmgr sync cluster
```

Comments:

- A linked cluster requires that each site has a repository disk. You must identify a repository disk for each site.
- A Repository disk resolves on the first node that the **clmgr** command is able to communicate with. For linked clusters, the first node that is defined for each site is the node that the **clmgr** command attempts to communicate with. In this example, the hdisk5 repository disk resolves on the SAP-A1 node and the hdisk11 repository disk resolves on the SAP-B1 node.
- You can specify the repository disk in PVID or UUID format.

Example: Create a resource group

Details:

This resource group will be a standard (non-concurrent) resource group, using default policies, and will be named db2RG. The resource group will contain a service IP address named access1, and an application controller named db2Controller. Further, the resource group will also manage two volume groups named vg1 and vg2, neither of which are concurrent.

Examples:

-
```
clmgr add resource_group db2RG SERVICE_IP=access1 \
      APPLICATIONS=db2Controller \
      VOLUME_GROUP=vg1,vg2
```
-
```
clmgr sync cluster
```

Example: Check current status

Details:

Very often it is important to know exactly what state a given object is in, so that appropriate actions can be taken. Using clmgr, this can be done via the query action.

Examples:

-
```
clmgr -a STATE query cluster
```
-
```
clmgr -a STATE query site siteA
```
-
```
clmgr -a STATE query node nodeA
```
-
```
clmgr -a STATE query resource_group rg1
```

Comments:

- For both the site and cluster classes, the STATE that is returned is a logical, worst-case aggregation of the member nodes. For example, in four node cluster, if even one node is experiencing an error, the status of the whole cluster will be reported as ERROR.

- The value returned will be in the standard ATTR=VALUE format, such as STATE=OFFLINE. If you need just the value, then you can combine a couple of other flags with the **-a** to good effect to achieve this. Using the flag combination of **-cSa** will return just the VALUE, such as OFFLINE. This will only work for a single value at a time.
- It is possible to retrieve multiple attributes at once with the **-a** flag, such as **-a NAME,STATE**. Further, the **-a** flag is not case sensitive (-a Name,state), and supports wildcards (-a N*).

Example: View all attributes and settings

Details:

PowerHA SystemMirror is a product that, once set up and fully tested, is typically no longer actively interacted with until either a problem occurs, or some sort of maintenance is required. When such things occur, it is necessary to be able to view the contents of the cluster, plus all settings. With `clmgr`, this is done using the query action, optionally requesting specific formats, colon-delimited or XML. The following command examples use resource groups, but the principles are the same for all object classes.

Examples:

- `clmgr query resource_group`
- `clmgr query resource_group rg1,rg2`
- `clmgr -c query resource_group rg1,rg2`
- `clmgr -x query resource_group rg1,rg2`
- `clmgr -v query resource_group`
- `clmgr -cv query resource_group`
- `clmgr -xv query resource_group`

Comments:

- When no target object is provided in a query command, and the verbose flag, **-v**, is not used, a simple listing of objects is displayed.
- When one or more target objects are provided in a query command, then all the known attributes or settings for those objects are displayed. This overrides the **-v** flag.
- When the **-v** flag is used with the query command, all the known attributes or settings for all known objects of the specified class are displayed.
- When detailed attributes or settings are displayed, by default they are displayed in ATTR=VALUE format, one per line. If **-c** is provided, then all values are displayed on one line in colon-delimited format. If **-x** is provided, then all attributes and values are displayed in a simple XML format.

Example: Display objects based on some filter or criteria

Details:

It is not uncommon to have large numbers of objects defined for a given class, such as resource groups, or to have large numbers settings defined within a given class. This can sometimes make it challenging to find the information that you really need. Fortunately, `clmgr` provides the ability to specify filtering criteria to the query action to solve this problem.

Examples:

- `clmgr query file_collection FILE="*rhosts*"`
- `clmgr query resource_group CURRENT_NODE=`get_local_nodename``

Comments:

- The first example shows a simple way to find an object that contains a particular value or setting; in this case, which file collection that contains a file named rhosts (note that wildcard characters are supported here).

- The second example shows a nice practical example of how to find an object that matches dynamic value. In this case, the example shows how to obtain the list of all resource groups that are currently running on the local node.
- This filtering capability can be used in combination with the **-a** flag to provide very powerful, flexible data retrieval.

Example: Make clmgr a little easier to use

Details:

Nothing in `clmgr` is case sensitive, which helps eliminate frustrating typing mistakes. Further, all actions, classes, and attributes or options can be shortened to either an explicitly named alias (such as start instead of online, or rg instead of resource_group), or to the fewest number of letters that make them unique. The following pairs of commands are functionally identical.

Examples:

- ```
  clmgr query resource_group
  clmgr q rg
  ```
- ```
  clmgr modify node mynode PERSISTENT_IP=myIP NETWORK=myNet
  clmgr mod node mynode pe=myIP netw=net_ether_0
  ```
- ```
  clmgr online node nodeA
  clmgr start node nodeA
  ```

Comments:

The shortening of actions and classes is intended for when `clmgr` is being used interactively within a terminal. Although these abbreviations can also be used in scripts, it is strongly suggested that scripts use the full names of both actions and classes. Doing so will provide more readable and serviceable code.

Example: Get instant help for clmgr

Details:

Help is always available online for `clmgr`. However, launching a web browser is often inconvenient, and sometimes impractical, or even impossible. So `clmgr` provides as much built-in help has it can, so that you might be able to get the help you need now. One type of help provided is when an object or value from a known set of objects or values is required. If an invalid object or value is provided, not only is an appropriate error message displayed, but also a list of the objects or values that are valid for that operation. This is wonderful in helping you overcome persistent typing errors! More help is available from `clmgr` when you are not sure what action, class, or object is needed. Just type as much as you know, then `clmgr` will tell you all the values that could possibly be next. Then you only have to choose one of them to proceed! Try running the following commands to see some examples of the help that `clmgr` is prepared to provide to you.

Examples:

- ```
  clmgr
  ```
- ```
  clmgr view
  ```
- ```
  clmgr view report
  ```
- ```
  clmgr view report -h
  ```

Comments:

The **-h** flag, when provided on the command line after either an object class or some set of option pairs, requests a listing of all valid options for this particular operation. This is the only flag in **clmgr** command that does not have to be positioned immediately after **clmgr** command itself.

Related information:
  ```

clpasswd command

Purpose

Change the current users password on all nodes in a cluster, or in a resource group.

Syntax

```
clpasswd [-g resource group] user
```

Description

The Cluster Password (**clpasswd**) utility lets users to change their own password on all nodes in a cluster, or in a resource group as specified by the PowerHA SystemMirror administrator, from a single node. Before users can change their password across cluster nodes, the PowerHA SystemMirror administrator adds any users who do not have root privileges to the list of users allowed to change their password.

This Cluster Password utility can also replace the AIX password utility from the SMIT fastpath **cl_passwd**.

The following table shows where a user's password is changed based on the user's authorization and the password utility that is active:

User authorization level	When the system password utility is linked to clpasswd and /bin/passwd is invoked	When the system password utility is active
User authorized to change password across cluster	The password is changed on all cluster nodes,	The password is changed on all cluster nodes.
***User not* authorized to change password across cluster**	The password is changed only on the local node.	The password is not changed.

Flags

-g Specifies the name of the resource group in which the user can change their password. The password is changed on each node in the specified resource group.

user
The username of the user who is changing their password.

Example

```
clpasswd -g rg1 myusername
```

clRGinfo command

Purpose

Creates a report that displays the location and state of one or more specified resource groups.

Syntax

```
clRGinfo [-h][-v][-s|-c][-t][-p][-a][-m][resgroup1] [resgroup2]...
```

Description

If the cluster services are not running on the local node, this **clRGinfo** command identifies a node where the cluster services are active and obtains the resource group information from the active cluster manager. If this command is used without any resource groups that are specified, the information about all the configured resource groups is displayed.

The output of the command displays both the global state of the resource group and the special state of the resource group on the local node.

The primary instance of a resource group can be in one of the following states:

Online
All resources for this resource group are active.

Error An error occurred while PowerHA SystemMirror was processing the resource group.

Unmanaged
Cluster services were stopped with the unmanage option.

Offline
The resource group is not active.

A resource group can be in the following transitional state while cluster events are in progress:

Acquiring
Resources for the resource group are being activated.

Releasing
Resources for the resource group are being released.

Temp error
A recoverable error occurred.

When a cluster uses sites and replicated resources, the resource group that contains the replicated resources has a primary and secondary instance that manages the replication end points. The **clRGinfo** command displays the following states for the secondary instance of a resource group:

Online secondary
All secondary resources for this resource group are active.

Error secondary
An error occurred while PowerHA SystemMirror was processing the secondary resources for a resource group.

Unmanaged secondary
Cluster services were stopped with the unmanage option.

Offline secondary
The secondary instance of a resource group is not active.

Acquiring secondary
The secondary resources for the resource group are being activated.

Releasing secondary
The secondary resources for the resource group are being released.

Temp error secondary
A recoverable error occurred while PowerHA SystemMirror was processing the secondary resources for a resource group.

Resource groups can be configured with dependencies that enable automatic placement and management of resource groups in relationship to other resource groups. The **clRGinfo** command displays the following states for resource groups with parent and child relationships and for resource groups that have location dependencies:

Offline due to parent offline
The child resource group is not active because the parent resource group is not active.

Offline due to fallover
A fallover occurred and the resource group is not active.

Offline due to lack of node
The resource group is not identified to a node in the cluster.

Offline due to target offline
The resource group that is involved in the relationship with a resource group is not active, and the configured dependencies dictate that this resource group must not be active.

Flags

-a Displays the current location of a resource group and its destination after a cluster event. Use this flag in pre-event and post-event scripts, especially in PowerHA SystemMirror clusters that have dependent resource groups. When PowerHA SystemMirror processes dependent resource groups, multiple resource groups can be moved at once with the **rg_move** event.

-c Displays the output in a colon-separated format.

-h Displays the usage message.

-m Displays the status of the application.

-p Displays the priority override location information for a resource group.

-s Displays the output in a colon-separated format.

-t Displays the delayed timer information, all delayed fallback timers, and settling timers that are currently active on the local node.

Note: You can only use this flag if the cluster manager is active on the local node.

-v Displays the verbose output.

Examples

1. The following example displays the report for running the **clRGinfo** command without specifying any flag parameters:

```
# clRGinfo
-----------------------------------------------------------------------------
Group Name     State                          Node
-----------------------------------------------------------------------------
VS_DATA_RG     ONLINE                         powerha53
               ONLINE                         powerha54
               ONLINE                         powerha63
               ONLINE                         powerha64

VS_REDO_RG     ONLINE                         powerha53
               ONLINE                         powerha54
               ONLINE                         powerha63
               ONLINE                         powerha64

RG1            ONLINE                         powerha53
               OFFLINE                        powerha54
               ACQUIRING                      powerha63
               OFFLINE                        powerha64
```

2. The following example displays the report for running the **clRGinfo** command in a cluster with sites.

```
# clRGinfo
-----------------------------------------------------------------------------
Group Name     State                      Node
-----------------------------------------------------------------------------
OASTRG         ONLINE                     als018022@site1
               ONLINE SECONDARY           alm194200@site2

VOTERG         ONLINE                     als018022@site1
               ONLINE SECONDARY           alm194200@site2
```

3. The following example displays the report for running the **clRGinfo -m** command:

```
$ /usr/es/sbin/cluster/utilities/clRGinfo -m

Group Name     State          Application state        Node
-----------------------------------------------------------------
Group1         ONLINE         ONLINE MONITORED         merry

Application state could be any one of the below possible values:
    OFFLINE
    ONLINE FAILED
    ONLINE FAILOVER
    ONLINE MONITORED
    ONLINE NOT MONITORED
    ONLINE MONITOR FAILED
    ONLINE MONITOR SUSPENDED
```

clRGmove command

Purpose

Perform a user-requested rg_move event to bring a resource group offline or online, or to move a resource group from one node to another node.

Syntax

```
clRGmove -g <groupname> -n <nodename> | -x -n <sitename> |-r | -a [-m | -u | -d] [-i] [-s true | false]
```

Description

You can use the **clRGmove** to manually control the location and state of resource groups.

You can perform any of the following actions for a non-concurrent resource group:

- Take the resource group offline from an online or online secondary node.
- Bring the resource group online or online secondary to a specific node.
- Move the resource group from its current hosting node to a new location.

You can perform any of the following actions for a concurrent resource group:

- Take the resource group offline from all nodes in the group node list.
- Take the resource group offline from one node in the group node list.
- Bring the resource group online on all nodes in the group node list.
- Bring the resource group online on one node in the group node list.

Priority Override Location

A priority override location overrides all other node policies and possible locations for the resource group.

The following are the priority override locations for non-concurrent resource groups:

- For every non-concurrent resource group movement that uses the `-n` flag to explicitly specify a destination instead of the `-r` flag, the destination becomes the priority override location. The priority override location lasts until you explicitly use the `-r` flag for the location instead of the `-n` flag when you manually move the resource group again.
- When you move a resource group offline, the resource group remains offline until you manually bring it back online. If you manually bring the resource group back online with the `-n` flag to specify a node, that node becomes the priority override location. When you bring a resource group back online with the `-r` flag, the active highest priority node is used and the priority override location is removed from the resource group.

The following are the priority override locations for concurrent resource groups:

- When you bring a concurrent resource group offline on all nodes, the priority override location is in the *OFFLINE* state for all nodes in the resource group. When you bring a concurrent resource group offline on just one node, the *OFFLINE* state for the resource group on the node is added to the priority override location list.
- When you bring a concurrent resource group online on all nodes, the priority override location is removed for all nodes in the resource group. When you bring a concurrent resource group online on just one node, the *OFFLINE* state for the resource group on that node is removed from the priority override location list.

For all resource group movements you can use one of the following movements:

non-persistent movement
Lasts until all nodes in the cluster are offline. As soon as the entire cluster goes offline, the priority override location is forgotten, and the resource group resumes normal behavior when the cluster comes back online.

persistent movement
Lasts after a cluster reboots. The priority override location remains when the cluster comes back online.

Limitations

The following are limitations for the **clRGmove** command:

- You can bring only one resource group online or offline at a time.
- When you move multiple resource groups with the command line, you must make sure that the request is rational. Thus, it is recommended that you use the SMIT interface for moving resource groups as it eliminates any possibility of administrative errors. To move resource groups with the SMIT interface enter `smit cspoc` from the command line and select **Resource Group and Applications**.

Flags

-a You can use this flag only for concurrent resource groups. Use this flag to bring the resource group online or offline for all nodes in the resource group. Use the `-n` flag to bring a concurrent resource group online or offline on a single node.

-d Brings the resource group offline. You cannot use this flag with the `-u` flag or the `-m` flag.

-g Specifies the name of the resource group to move in the following formats:

-g <groupname>
Specific a single resource group name.

-g "groupname1,groupname2,..."
Specifies a comma-separated list of multiple resource group names.

-i Runs the **clRGinfo** command after the resource group has been moved successfully.

-m Moves the resource group to another node. You cannot use this flag with the -u flag or the -d flag. Use this flag to move multiple online resource groups to another node one node at a time.

-n <nodename>
The name of the node that contains the resource group that is moved, brought online, or brought offline. You cannot use this flag with the -r or the -a flag. If the node name has a * character in front of it, that node is configured to be the highest priority node for this resource group and the resource group was moved to another node. If you move a resource group in a node that is identified with a * character, the movement changes the original configuration for the resource group.

-n <sitename>
The name of the site that contains the resource group that is moved across a site. You must use this flag with the -x flag. If the site name has a * character in front of it, that site is configured to be the highest priority site for this resource group and the resource group was moved to another site. If you move a resource group in a site that is identified with a * character, the movement changes the original configuration for the resource group.

-r You can use this flag only for non-concurrent resource groups. Use the highest priority node that is available for the destination node where the resource group is moving. This flag removes the priority override location attribute for the resource group that is being moved. You can use this flag only when you are bringing a non-concurrent resource group online or moving a non-concurrent resource group to another node. You cannot use this flag with the -n flag or the -a flag.

-s true | false
Specifies actions on the primary or secondary instance of a resource group (if sites are defined). You use this flag to take the primary or the secondary instance of the resource group offline, online, or move it to another node within the same site. You can use this flag with the -r, -d, -u, and -m flags.

-s true
Specifies actions on the secondary instance of a resource group.

-s flase
Specifies actions on the primary instance of a resource group.

-u Brings the resource group online. You cannot use this flag with the -d flag or the -m flag.

-x
You can use this flag to move the resource group across a site. You must use this flag with the -n <sitename> flag.

Examples

1. To bring an offline non-concurrent resource group online on a node named nodeB:
   ```
   clRGmove -g rgA -n nodeB -u
   ```
2. To move an online non-concurrent resource group to another node named nodeB:
   ```
   clRGmove -g rgA -n nodeB -m
   ```
3. To move multiple online non-concurrent resource groups to another node named nodeB:
   ```
   clRGmove -g "rgA,rgB,rgC" -n nodeB -m
   ```
4. To bring an online non-concurrent resource group offline on a node named nodeB:
   ```
   clRGmove -g rgA -n nodeB -d
   ```
5. To move an online non-concurrent resource group to the active highest priority node that is removing the previous configuration settings that are caused by another rg_move event:
   ```
   clRGmove -g *rgA -m -r
   ```
6. To bring an online concurrent resource group offline on one node named nodeB:
   ```
   clRGmove -g rgA -n nodeB -d
   ```
7. To bring an online concurrent resource group offline on all nodes:
   ```
   clRGmove -g rgA -a -d
   ```
8. To bring an offline concurrent resource group online on one node named nodeB:

```
clRGmove -g rgA -n nodeB -u
```

9. To bring an offline concurrent resource group online on all nodes:

```
clRGmove -g rgA -a -u
```

10. To move a resource group to a site named `site2`:

```
clRGmove -s false -x -g rgA -n site2
```

Related reference:

"clmgr command" on page 40

clruncmd command

Purpose

Restores cluster manager to normal operation.

Syntax

```
clruncmd nodename
```

Note: The `nodename` represents the name of a cluster node where cluster services are active.

Description

The **clruncmd** command instructs the cluster manager on the specified node to resume event processing after an event script failure occurs. Run the **clruncmd** command only after the reasons for the failure have been manually corrected. After a event script failure occurs, the remainder of the failed event is skipped, and the event processing resumes with the next event in the event queue. You must manually perform any actions which were skipped after the event failure occurred.

Example

To instruct the cluster manager to return to normal operations for a node that is named `node1`, enter:

```
clruncmd node1
```

Related reference:

"clmgr command" on page 40

clshowres command

Purpose

Displays resource group information for a cluster or a node.

Syntax

```
clshowres [-g group ] [-n nodename ] [-d odmdir ]
```

Flags

-g group
Name of resource group to show.

-n nodename
Searches the resources Configuration Database from the specified node.

-d odmdir
Specifies *odmdir* as the ODM object repository directory instead of the default **/etc/objrepos**.

Examples

1. Run the following command to list all the resource group information for the cluster.
   ```
   clshowres
   ```
2. Run the following command to lists the resource group information for clam node.
   ```
   clshowres -n clam
   ```

---

clshowsrv command

Purpose

Displays the status of PowerHA SystemMirror subsystems.

Syntax

```
clshowsrv { -a | -v | subsystem ...}
```

Description

The **clshowsrv** command displays the status of PowerHA SystemMirror subsystems. Status includes the subsystem name, group name, process ID, and status. The status of a daemon can be any of the states that are reflected by the System Resource Controller (SRC) subsystem (active, inoperative, warned to stop, and so on).

Flags

-a Displays all the PowerHA SystemMirrordaemons.

subsystem
Displays the status of the specified PowerHA SystemMirror subsystem. Valid values for this flag are `clstrmgrES`, `clinfoES`, and `clcomd`. If you specify more than one subsystem, you must separate the entries with a space.

-v Displays all RSCT, PowerHA SystemMirror, and optional PowerHA SystemMirror daemons.

Examples

1. To display the status of all PowerHA SystemMirror and RSCT subsystems, enter:
   ```
   clshowsrv -v
   ```

   The command displays the output information similar to the following example:
   ```
   Local node: "hadev11" ("hadev11.aus.stglabs.ibm.com", "hadev11.aus.stglabs.ibm.com")
            Cluster services status:   "OFFLINE" ("ST_INIT")
            Remote communications:     "UP"
            Cluster-Aware AIX status:  "UP"

   Remote node: "hadev12" ("hadev12.aus.stglabs.ibm.com", "hadev12")
            Cluster services status:   "OFFLINE" ("ST_INIT")
            Remote communications:     "UP"
            Cluster-Aware AIX status:  "UP"

   Status of the RSCT subsystems used by PowerHA SystemMirror:
   Subsystem         Group            PID          Status
    cthags           cthags           9371848      active
    ctrmc            rsct             11862036     active

   Status of the PowerHA SystemMirror subsystems:
   Subsystem         Group            PID          Status
    clstrmgrES       cluster          12124406     active

   Status of the CAA subsystems:
   ```

```
Subsystem         Group            PID          Status
 clconfd          caa              10420354     active
 clcomd           caa              8912916      active
```

2. To display the status of all PowerHA SystemMirror subsystems, enter:
   ```
   clshowsrv -a
   ```
3. To display the status of the `clstrmgr` subsystem, enter:
   ```
   clshowsrv clstrmgrES
   ```
4. To display the status of the `clstrmgr` and `clinfo` subsystems, enter:
   ```
   clshowsrv clstrmgrES clinfo
   ```

Related reference:
"clmgr command" on page 40

---

clsnapshot command

Purpose

Creates a cluster snapshot. A snapshot is a set of ASCII files which contain PowerHA SystemMirror cluster configuration data and state information.

Syntax

```
clsnapshot [-a] [-c] [-C] [-d description] [-e] [-f true|false] [-g] [-h]
[-i] [-l] [-m methodlist] -n filename [-N filename] [-o odmdir]
[-q] [-r] [-R] [s] [-t]
```

Description

The **clsnapshot** command creates, modifies, or removes two files. The first file is identified by the file extension .odm and contains the current PowerHA SystemMirror ODM class objects. You can write a brief description to the file. The second file with an extension of .info contains information useful for troubleshooting PowerHA SystemMirror clusters.

The **clsnapshot** command is run on every configured node to obtain node-specific information.

You can use the **clsnapshot** command to apply a snapshot to the current cluster hardware. A verification utility is run, and must pass before the configuration information is synchronized to the cluster nodes. You can use the `-f` flag to forces a snapshot to be applied even if the verification routine fails.

Note: The environment variable *SNAPSHOTPATH* contains the path that leads to the snapshot file. By default, this path is `/usr/es/sbin/cluster/snapshots`.

Flags

-a Apply a cluster snapshot

-c Create a cluster snapshot

-C Do not refresh an active cluster resource when you are applying a snapshot.

-d text
Added a description to the snapshot.

-e Save cluster logs in the snapshot. Saving the logs to the snapshot can significantly increase the file size of the snapshot.

-f true|false
Force the application of a snapshot if verification fails.

-g Generate a temporary ODM holding the snapshot

-h Usage of the snapshot

-i Generates files with the .info extension.

-l Lists the snapshot files.

-m methodlist
Runs each custom snapshot method that is listed in the methodlist file.

-n file
Specifies the name of the snapshot.

-N file
Specifies the new name of the snapshot.

-o odmdir
Specify the ODM directory (ODMDIR) for the PowerHA SystemMirror ODM classes.

-r Removes a snapshot.

-R Replaces a snapshot.

-s Displays a snapshot.

-t Resets cluster options.

Related reference:
"clmgr command" on page 40

---

clsnapshotinfo command

Purpose

Retrieves and displays certain PowerHA SystemMirror cluster configuration information.

Syntax

```
clsnapshotinfo [-m <METHOD> [<METHOD#2> ...]]
```

Description

The **clsnapshotinfo** command runs PowerHA SystemMirror and AIX commands to gather information about the PowerHA SystemMirror cluster. The **clsnapshotinfo** command gathers information from only the node where the command is run. The output from the command is written to STDOUT. When the **clsnapshotinfo** command is run from the **clsnapshot** command, which happens automatically, information from all nodes in the cluster is gathered and the output is stored in a snapshot file with the .info extension.

It is recommended that you run the **clsnapshotinfo** command as part of the **clsnapshot** command to collect as much information as possible about the cluster.

Flags

-m Specifies one or more custom snapshot methods. The output from these methods is part of the overall data that is collected by the **clsnapshotinfo** command.

Related reference:
"clmgr command" on page 40

clstat command (ASCII mode and X Windows mode)

Note: This topic contains information about the ASCII mode and the X Windows mode for the **clstat** command.

ASCII mode

Purpose

Cluster Status Monitor (ASCII mode).

Syntax

```
clstat [-c cluster ID | -n cluster name] [-i] [-r seconds] [-a] [-o][-s]
```

Flags

-c cluster id
Displays cluster information only about the cluster with the specified ID. If the specified cluster is not available, clstat continues looking for the cluster until the cluster is found or the program is canceled. May not be specified if the -i option is used.

-i Runs ASCII **clstat** in interactive mode. Initially displays a list of all clusters accessible to the system. The user must select the cluster for which to display the detailed information. A number of functions are available from the detailed display.

-n name
Displays cluster information about the cluster with the specified name. May not be specified if the -i option is used.

-r seconds
Updates the cluster status display at the specified number of seconds. The default is 1 second; however, the display is updated only if the cluster state changes.

-a Causes **clstat** to display in ASCII mode.

-o Provides a single snapshot of the cluster state and exits. This flag can be used to run **clstat** out of a **cron** job. Must be run with **-a** ; ignores **-i** and **-r** options.

-s Displays service labels and their state (up or down).

X Windows mode

Purpose

Cluster Status Monitor (X Windows mode).

Syntax

```
clstat [-a] [-c id | -n name ] [-r tenths-of-seconds ][-s]
```

Flags

-a Runs **clstat** in ASCII mode.

-c id
Displays cluster information only about the cluster with the specified ID. If the specified cluster is not available, **clstat** continues looking for the cluster until the cluster is found or the program is canceled. Might not be specified if the **-n** option is used.

-n name
Displays cluster information only about the cluster with the specified name.

-r tenths-of-seconds
The interval at which the **clstat** utility updates the display. For the graphical interface, this value is interpreted in tenths of seconds. By default, **clstat** updates the display every 0.10 seconds.

-s Displays service labels and their state (up or down).

Examples

1. Run the following command to display the cluster information about the mycluster cluster.
   ```
   clstat -n mycluster
   ```
2. Runs ASCII clstat in interactive mode, allowing multi-cluster monitoring.
   ```
   clstat -i
   ```
   The following are the buttons on X Window System Display:

   Prev Displays previous cluster.

   Next Displays next cluster.

   Name:Id
   Refresh bar, pressing bar causes **clstat** to refresh immediately.

   Quit Exits application.

   Help Pop-up help window shows the **clstat** manual page.

---

clstop command

Purpose

Stops the cluster subsystems.

Syntax

```
clstop { -f | -g | -gr } [-s] [-y] [ -N | -R | -B ]
```

Description

The **clclstop** stops cluster services on the local node and processes any active resource groups according to the flags that you specify. The command optionally removes automatic start on reboot through the entry in the `/etc/inittab` file.

Flags

-f Forces a shutdown. Cluster daemons terminate without running any local procedures.

-g Graceful shutdown with no takeover.

-gr
Graceful shutdown with the resources that are being released by this node and taken over by another node. The daemon terminates gracefully, and the node releases its resources, which are taken over. A node list must be specified for graceful shutdown with takeover.

-s Performs a silent shutdown. This flag does not broadcast a shutdown message through **wall** command. The default setting is to broadcast.

-y Do not ask operator for confirmation before shutting down the cluster nodes. This flag is the default.

-B Stop now and on subsequent system restart.

-N Shut down now.

-R Stops on subsequent system restart and removes the entry in the `/etc/inittab` file.

Note: The `/etc/rc.shutdown` file is an optional file that contains commands that are run during the shutdown command.

Examples

1. To shut down the cluster node by using the gracefully option and releasing the resources without sending a warning message to users before the cluster processes are stopped, enter:

   ```
   clstop -gr -s -y
   ```

2. To forcefully and immediately shut down the cluster on all cluster nodes (resources not released) with a warning message that is broadcast to users before the cluster processes are stopped, enter:

   ```
   clstop -f -y
   ```

3. To shut down the cluster node by using the gracefully option and releasing resources that are taken over with a warning message that is broadcast to users before cluster processes are stopped, enter:

   ```
   clstop -gr -y
   ```

Related reference:

"clmgr command" on page 40

---

cltopinfo command

Purpose

Displays complete topology information: The cluster name, total number of networks, total number of missed heartbeats and nodes configured in the cluster. Displays all the configured networks for each node. Displays all the configured interfaces for each network. Also displays all the resource groups defined.

Syntax

```
cltopinfo [-c] [-i] [-n] [-w]
```

Flags

-c Shows the cluster name and the security mode (Standard or Enhanced)

-i Shows all interfaces configured in the cluster. The information includes the interface label, the network it's attached to (if appropriate), the IP address, netmask, nodename and the device name.

-n Shows all the nodes configured in the cluster. For each node, lists all the networks defined. For each network, lists all the interfaces defined and the distribution preference for service IP label aliases (if defined).

-w Shows all the networks configured in the cluster. For each network, lists all the nodes attached to that network. For each node, lists all the interfaces defined and the distribution preference for service IP label aliases (if defined).

Example 1

To show all of the nodes and networks defined in the cluster (nodes coffey1 and lee1), use the **cltopinfo** command. The following cluster is configured with IPv4 addresses and IPv6 addresses. The output looks similar to the following:

```
Cluster Name: hacmp_full_ipv6
Cluster Connection Authentication Mode: Standard
Cluster Message Authentication Mode: None
Cluster Message Encryption: None
Use Persistent Labels for Communication: No
There are 2 node(s) and 2 network(s) defined

NODE coffey1:
   Network net_ether_01
```

```
      service_ipv4_2   1.8.4.2
      service_ipv6_1   fe80::c862:67ff:fe58:5646
      coffey1_boot3   1.4.6.4
      coffey1_boot1   1.2.4.4
   Network net_ether_02
      service_ipv4_32   1.8.4.4
      service_ipv6_31  fe80::c862:67ff:fe58:5846
     coffey1_boot_v6  fe80::c872:67ff:fe59:8647
     coffey1_boot_v6  fe80::c872:678f:fe95:8683
NODE lee1:
   Network net_ether_01
      service_ipv4_2   1.8.4.2
      service_ipv6_1   fe80::c862:67ff:fe58:5646
      lee1_boot1   1.2.4.3
      lee1_boot3   1.4.6.3
   Network net_ether_02
      service_ipv4_32   1.8.4.4
      service_ipv6_31   fe80::c862:67ff:fe58:5846
      lee1_boot_v6   fe80::c672:fe56:fe82:2345
      lee1_boot_v6   fe80::fe34:3456:f873:f345

Resource Group RG1
   Startup Policy    Online On Home Node Only
   Fallover Policy    Fallover To Next Priority Node In The List
   Fallback Policy    Fallback To Higher Priority Node In The List
   Participating Nodes    coffey1 lee1
   Service IP Label       service_ipv4_1
   Service IP Label       service_ipv4_31

Resource Group RG2
   Startup Policy    Online On Home Node Only
   Fallover Policy    Fallover To Next Priority Node In The List
   Fallback Policy    Fallback To Higher Priority Node In The List
   Participating Nodes    lee1 coffey1
   Service IP Label       service_ipv4_2
   Service IP Label       service_ipv4_32
```

Example 2

To show the cluster name and current security mode, use the **cltopinfo** command. The output looks similar to the following:

```
# cltopinfo -c

 Cluster Name: c10
 Cluster Connection Authentication Mode: Standard
 Cluster Message Authentication Mode: None
 Cluster Message Encryption: None
 Use Persistent Labels for Communication: No
```

Example 3

To show all of the nodes defined in the cluster, use the **cltopinfo** command. The following cluster is configured with IPv4 addresses and IPv6 addresses. The output looks similar to the following:

```
# cltopinfo -n

 NODE abby:
      Network net_ether_01
    abby_en1stby    192.168.121.7
    abby_en0boot    192.168.120.7
      Network net_ether_02
abby_boot1_v6  fe80::c872:67ff:fe59:8647
abby_boot2_v6  fe80::c872:678f:fe95:8683
      Network net_rs232_01
      Network net_rs232_02
```

```
    abby_tty0_01    /dev/tty0

 NODE polly:
      Network net_ether_01
    polly_en0boot    192.168.120.9
    polly_en1stby    192.168.121.9
    polly_en2boot    192.168.122.9
      Network net_ether_02
polly_boot1_v6  fe80::c672:fe56:fe82:2345
polly_boot2_v6  fe80::fe34:3456:f873:f345
      Network net_rs232_01
      Network net_rs232_02
    polly_tty0_01    /dev/tty0
```

Example 4

To show all of the networks defined in the cluster, use the **cltopinfo** command. The following cluster is configured with IPv4 addresses and IPv6 addresses. The output looks similar to the following:

```
# cltopinfo -w

 Network net_ether_01
      NODE abby:
    abby_en1stby    192.168.121.7
    abby_en0boot    192.168.120.7
      NODE polly:
    polly_en0boot    192.168.120.9
    polly_en1stby    192.168.121.9
    polly_en2boot    192.168.122.9

 Network net_ether_02
      NODE abby:
abby_boot1_v6  fe80::c872:67ff:fe59:8647
abby_boot2_v6  fe80::c872:678f:fe95:8683
      NODE polly:
polly_boot1_v6  fe80::c672:fe56:fe82:2345
polly_boot2_v6  fe80::fe34:3456:f873:f345

 Network net_rs232_01
      NODE abby:
      NODE polly:

 Network net_rs232_02
      NODE abby:
    abby_tty0_01    /dev/tty0
      NODE polly:
    polly_tty0_01    /dev/tty0
```

Example 5

To show all of the interfaces defined in the cluster, use the **cltopinfo** command. The output looks similar to the following:

```
# cltopinfo -i
IP Label NetworkType   Node  Address If   Netmask   Pefixlenth
==================   ====  ======= ==== =======   ==========
abby_en1stby    net_ether_01  ether  abby  192.168.121.7  en2  255.255.255.0
abby_en0boot    net_ether_01  ether  abby  192.168.120.7  en1  255.255.255.0
abby_boot1_v6   net_ether_02  ether  abby  fe80::c872     en3           64
abby_boot2_v6   net_ether_02  ether  abby  fe80::c672     en4           64
abby_tty0_01    net_rs232_02  rs232  abby  /dev/tty0      tty0
polly_en0boot   net_ether_01  ether  polly 192.168.120.9  en1  255.255.255.0
polly_en1stby   net_ether_01  ether  polly 192.168.121.9  en2  255.255.255.0
polly_en2boot   net_ether_01  ether  polly 192.168.122.9  en3  255.255.255.0
```

```
polly_boot1_v6  net_ether_02  ether  polly fe80::c072     en4           64
polly_boot2_v6  net_ether_02  ether  polly fe80::c172     en5           64
polly_tty0_01   net_rs232_02  rs232  polly /dev/tty0      tty0
```

clvaryonvg command

Purpose

Varies on a volume group.

Syntax

```
clvaryonvg [-F] [-f] [-n] [-p] [-s] [-o] <vg>
```

Description

The **clvaryonvg** command is designed as a replacement for the **varyonvg** command that is part of the AIX operating system. This command performs some checks on the volume group to determine if there are any changes were made to the volume group before calling the AIX **varyonvg** command. If any changes have been made since the last time the volume group was varied on locally, the volume group is exported and then imported before it is varied on. This process verifies that all nodes have a consistent view of the content for the volume group.

If a system failure occurs during a volume group update, the volume group can become invisible to the node. The following mechanisms are implemented to safeguard against a system failure:

- Before you export a volume group, a file is created in the `/usr/es/sbin/cluster/etc/vg` directory that is called `<VG>.replay`, where *VG* is the name of the volume group. This file is a shell script that contains a set of commands to restore the volume group if it becomes invisible to the node or does not exist. If the volume group does not exist, the commands in the `<VG>.replay` file are automatically run the next time that you use the **clvaryonvg** command.
- If the replay file does not fix the problem, you can view the messages in the `hacmp.out` file. These messages explain how to manually restore the volume group. The messages in the `hacmp.out` file can also be found in the `/usr/es/sbin/cluster/etc/vg/<VG>.desc` file, where *VG* is the name of the volume group. A copy of the failed replay file is placed in the `/var/tmp` directory.

Flags

-f Passes the flag to the **importvg** command or the **varyonvg** command.

-F Forces an update by using the **exportvg** command or the **importvg** command on the volume group and ignoring the time stamp.

-n Disables the synchronization of the stale physical partitions within the volume group. This flag is passed to the **varyonvg** command.

-p Specifies that all physical volumes must be available to use the **clvaryonvg** command.

-s Makes the volume group available only in system management mode.

-o Leaves the volume group varied off after completion. The completion process includes performing all the integrity checks, and if required, running the **importvg** command and the **exportvg** command.

Examples

1. To activate volume group that is labeled *vg03*, enter:
   ```
   clvaryonvg vg03
   ```
2. To force the update of a node's information on volume group that is labeled *vg03*, enter:
   ```
   clvaryonvg -F vg03
   ```

get_local_nodename command

Purpose

Retrieves the name of the local node.

Syntax

```
get_local_nodename
```

Description

Displays the name of the local node.

Example

To display the name for the local node, enter

```
get_local_nodename
```

halevel command

Purpose

Displays the PowerHA SystemMirror version, release, modification, and service pack level that is installed on your system.

Syntax

```
halevel [-h|-?] [-s] [-x]
```

Description

If you run this command from a PowerHA SystemMirror client the command does not work correctly. You must run this command from a PowerHA SystemMirror server node.

Flags

-h | -?
Displays help information.

-s Displays the service pack level.

-x Turns on debugging (ksh set -x)

Examples

1. To display the PowerHA SystemMirror version, release, and modification level, enter:
   ```
   halevel
   ```
2. To display the PowerHA SystemMirror version, release, modification, and service pack level, enter:
   ```
   halevel -s
   ```
3. To display the PowerHA SystemMirror version, release, modification, and service pack level on all cluster nodes, enter:
   ```
   /usr/es/sbin/cluster/cspoc/cli_on_cluster -S halevel -s
   ```

rc.cluster command

Purpose

Use the **rc.cluster** command to setup the operating system environment and start the cluster daemons across cluster nodes.

Note: Arguments associated with a particular flag must be specified immediately following the flag. PowerHA SystemMirror

Syntax

```
rc.cluster [-boot] [b] [-i | -I] [-N | -R | -B] [-M | -A] [-r] [-v] [-x] [-C interactive|yes]
```

Flags

-boot
Configures the service network interface to be on its boot address if IPAT is enabled.

-i Starts the Cluster Information (**clinfoES**) daemon with its default options.

-I Starts the Cluster Information (**clinfoES**) daemon with traps enabled.

-b Broadcasts the startup.

-N Starts the daemons immediately (no **inittab** file change).

-R Starts the PowerHA SystemMirror daemons on system restart only. The PowerHA SystemMirror startup command is added to the **inittab** file.

-B Starts the daemons immediately and adds the PowerHA SystemMirror entry to the **inittab** file.

-C Specifies the mode to use for corrective action when a problem occurs. Specify **yes** to automatically correct problems. Specify **interactive** to be prompted before each corrective action is run.

-M Starts the cluster services with Manual resource acquisition mode. Use this option if you want to bring the resource groups online manually.

-A Starts the cluster services with Automatic resource acquisition mode. Use this option if you want to bring resource groups online automatically on cluster startup. This is the default option.

-r Reacquires cluster resources after a forced down. Use this option if you changed the state of any cluster resources (ip labels, disks, applications) while the cluster was forced down.

-v Ignore verification errors during startup (auto ver sync)

-x Activates NFS cross-mounts.

Example

To start the cluster with **clinfo** services and to broadcast the event, run the following command:

```
rc.cluster -boot -N -i
```

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as the customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Index

C

R

Printed in USA