

IBM Cluster Systems Management for Linux



# Command and Technical Reference

*Version 1.3.2*



IBM Cluster Systems Management for Linux



# Command and Technical Reference

*Version 1.3.2*

**Note**

Before using this information and the product it supports, read the information in “Notices” on page 161.

**Second Edition (September 2003)**

This edition of the *IBM Cluster Systems Management for Linux Command and Technical Reference* applies to version 1, release 3, modification 2 of IBM Cluster Systems Management for Linux (CSM) licensed program, product number 5765-E88, CSM for Linux on pSeries, product number 5765-G16, and to all subsequent releases and modifications of this product until otherwise indicated in new editions. Significant changes or additions to the text and illustrations are indicated by a vertical line ( | ) to the left of the change.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation  
Department 55JA, Mail Station P384  
2455 South Road  
Poughkeepsie, NY 12601-5400  
United States of America

FAX (United States & Canada): 1+845+432-9405

FAX (Other Countries):

Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)

IBM Mail Exchange: USIB6TC9 at IBMMAIL

Internet e-mail: mhvrdfs@us.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2001, 2003. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>About this book</b>	v
Who should use this book	v
Typographic conventions	v
ISO 9000	v
Prerequisite and related information	vi
How to send your comments	vi
How to read command synopsis	vi
<b>CSM commands</b>	1
addlcsext Command	2
cfmupdatenode Command	5
chnode Command	9
chrconsolecfg Command	11
chsnmp Command	14
copycsmpkgs Command	17
csm	19
csmbackup Command	22
csmconfig Command	25
csmrestore Command	28
csmsetupks Command	30
csmsetupyast Command	35
csmsis2yast Command	39
csmstat Command	41
dcem Command	44
definenode Command	46
dsh Command	54
dshbak Command	62
getadapters Command	64
installms Command	68
installnode Command	71
kscfg.tmpl File	76
lshwinfo Command	78
lshwstat Command	81
lsnode Command	84
lssnmp Command	87
mgmtsvr Command	90
monitorinstall Command	92
nodeattributes	95
nodedef File	100
nodegrp Command	102
noderange	105
predefined-condresp Command	110
predefined-nodegroups Command	112
probemgr Command	114
rconsole Command	118
rconsolerefresh Command	122
reventlog Command	124
rmlcsext Command	126
rmnode Command	128
rpower Command	130
smsupdatenode Command	134
systemid Command	139
uninstallms Command	141

updateisvr Command . . . . .	143
updatenode Command . . . . .	145
<b>Appendix A. CSM log files . . . . .</b>	<b>149</b>
<b>Appendix B. Glossary . . . . .</b>	<b>151</b>
<b>Appendix C. Bibliography . . . . .</b>	<b>157</b>
Related information . . . . .	157
Obtaining publications . . . . .	157
Redbooks . . . . .	157
Other CSM information . . . . .	157
Getting XCAT tools . . . . .	158
Getting help from IBM . . . . .	158
Finding service information . . . . .	158
Calling IBM for help. . . . .	158
Contacting CSM development . . . . .	159
<b>Notices . . . . .</b>	<b>161</b>
Trademarks. . . . .	162
Publicly Available Software . . . . .	163
GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION. . . . .	166
<b>Index . . . . .</b>	<b>171</b>

---

## About this book

This book provides detailed information about the commands and files provided with IBM Cluster Systems Management for Linux (CSM). It is organized in alphabetical order, by command name.

CSM for Linux currently supports Linux nodes only; for information on using Linux and AIX 5L nodes in the same cluster, see the CSM for AIX 5L documentation.

### Attention!

- Except where noted, the information in this book also applies to Linux on pSeries clusters. Linux on pSeries clusters must be homogeneous – both a management server and Managed nodes running Linux on pSeries servers only.
- Refer to the Statement of Direction in the IBM Cluster Systems Management V1.3.2 Announcement Letter for information on support for the eServer 325.

---

## Who should use this book

This book is intended for users of CSM commands. CSM command users should:

- Be highly skilled in using most Linux commands and utilities.
- Possess a solid understanding of a Linux-based operating system.
- Be familiar with fundamental networking/distributed computing environment concepts.

---

## Typographic conventions

This book uses the following typographic conventions:

Convention	Usage
<b>bold</b>	<b>Bold</b> words or characters represent system elements that you must use literally, such as: command names, file names, flag names, and path names.
constant width	Examples and information that the system displays appear in constant-width typeface.
<i>italic</i>	<i>Italicized</i> words or characters represent variable values that you must supply. <i>Italics</i> are also used for book titles, for the first use of a glossary term, and for general emphasis in text.
[item]	Used to indicate optional items.
<Key>	Used to indicate keys you press.

---

## ISO 9000

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

---

## Prerequisite and related information

If you are using Red Hat Linux version 8.0, you must set your **LANG** environment variable to **C** for the man pages to format correctly.

See Appendix C, “Bibliography,” on page 157 for

- A list of related publications
- How to get help from IBM
- Information on Linux XCAT tools

---

## How to send your comments

Your feedback is important in helping us to produce accurate, high-quality information. If you have any comments about this book or any other CSM documentation:

- Send your comments by e-mail to: [mhvrcfs@us.ibm.com](mailto:mhvrcfs@us.ibm.com)  
Include the book title and order number, and, if applicable, the specific location of the information you have comments on (for example, a page number or a table number).
- Fill out one of the forms at the back of this book and return it by mail, by fax, or by giving it to an IBM representative.

To contact the IBM cluster development organization, send your comments by e-mail to: [cluster@us.ibm.com](mailto:cluster@us.ibm.com)

---

## How to read command synopsis

Synopsis statements represent command line synopsis, and consist of symbols such as brackets [ ], braces { }, and pipe characters |, for example:

```
mgmtsvr [-h] [-k] [-v | -V] [-N] [-n hostname] [host | -d]
```

Command synopsis statements in this book and in CSM man pages use the following conventions:

- Items that must be entered literally on the command line are in **bold**. These items include the command name, options, and literal characters.
- Items that represent variables that you must replace with a name are in *italics*. These items include parameters that follow options such as *Files* or *Directories*.
- Options and parameters enclosed in brackets are optional.
- Options and parameters enclosed in braces are required.
- Options and parameters not enclosed in either brackets or braces are required.
- Options preceded by two dashes -- are case-sensitive and must be specified in their entirety.
- The pipe character signifies that you choose one option or the other. For example, [a | b] indicates that you *can* choose either a or b, but not both. Similarly, {a | b} indicates that you *must* choose either a or b.
- Ellipses ... signify that you can repeat the parameter on the command line.
- The dash – represents standard output.



---

# CSM commands

---

## addlcsext Command

### Purpose

**addlcsext** - Adds IBM Director Linux Cluster Support (LCS) extension components.

### Synopsis

```
addlcsext [-h | --help] [-A | --agent] [-S | --server] [-p pkg_path[:pkg_path...]] [-v | --verbose] [-- restart]
```

### Description

The LCS extension is an add-on to IBM Director, which provides access to some CSM for Linux function through the IBM Director Console. The **addlcsext** command is used to add the necessary LCS extension components to Linux IBM Director Servers and CSM management servers and nodes.

**Note:** This command is not supported for Linux on pSeries clusters.

If run without the **-A** and **-S** options, the **addlcsext** command determines which extension components are needed and adds them. However, one or more of these options can be specified to limit which extension components are added.

The **addlcsext** command cannot be run on a CSM management server unless the **csmconfig** command has been run first to install a license key file.

This command is used on Linux Director Servers and on CSM management servers. On Linux Director Servers with no CSM code installed, **addlcsext** is run directly from the CSM CD-ROM. On Windows Director Servers, the **addlcsext.exe** install wizard is run directly from the CSM CD-ROM. On CSM management servers, **addlcsext** is run from **/opt/csm/bin**; the command is usually not run explicitly on CSM managed nodes. Once the command has been run on the CSM management server, all future use of the **installnode** and **updatenode** commands ensures that **addlcsext** gets run automatically on the specified nodes.

To install the LCS extension, the IBM Director Server or Agent must first be stopped if it is already running. For the extension to become active, the Server or Agent to which it has been added must then be started. Because of the potential for disruption, **addlcsext**, if run locally without the **-- restart** option, prompts you for permission to stop and restart the local Server or Agent. If you decline, **addlcsext** does not install the extension and exits.

**Note:** When **installnode** or **updatenode** cause **addlcsext** to run on a node, there is no prompt.

### Parameters

None.

### Options

#### **-A | -- agent**

Installs the IBM Director Agent if not already installed, and the Agent's LCS extension components. The basic IBM Director Agent code is found in the ITDAgent and DirAgent RPM packages, and the Agent's LCS extension components are found in the **csm.director.agent** RPM package.

**-h | --help**

Displays command usage information.

**-p *pkg\_path*...**

Specifies a colon-separated list of directories to search for the IBM Director Agent and LCS extension RPM™ packages. The default path is **../lcsext/packages**, relative to the directory that **addlcsext** is running in. If an explicit *pkg\_path* is specified, it is added to the front of the default *pkg\_path*.

**--restart**

Forces the IBM Director Agent or Server to restart without prompting.

**-S | --server**

Installs the IBM Director Server's LCS extension components. The Server's LCS extension components are found in the **esm.director.server** RPM package.

**-v | --verbose**

Specifies verbose mode.

## Exit Status

- 0** The command completed successfully.
- 1 - 4** The number of packages that did not install.
- 96** An IBM Director error occurred.
- 97** An RPM error occurred.
- 98** Prerequisites are missing.
- 99** Command line arguments are incorrect.

## Security

The command requires root access to the cluster management server.

## Examples

1. To let the **addlcsext** command determine which extension components to add, enter:  

```
addlcsext
```
2. To add only the LCS extension components that are needed by the IBM Director Agent, enter:  

```
addlcsext -A
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

**/opt/csm/bin/addlcsext**

**/lcsext/addlcsext** on CSM CD-ROM

**\\csext\\addlcsext.exe** – Windows install wizard on CSM CD-ROM

**addlcsex**

## **See Also**

The **csmconfig**, **installnode**, **rmlcsex**, and **updatenode** commands.

## **Author**

IBM CSM Development - cluster@us.ibm.com

## cfmupdatenode Command

### Purpose

**cfmupdatenode** - Distributes configuration files across a CSM cluster. The **cfmupdatenode** command is called by the **updatenode** command.

### Synopsis

```
cfmupdatenode [-h] [-v | -V] [-b] [[-y] | [-c]] [-q [-s]] [-r remote_shell_path] [-t
timeout] [-M max_child_process] [-d distfile_location] [-f filename] {-a | -N
nodegroups | --file filename | [-n] node_list}
```

### Description

The **cfmupdatenode** command provides the major function of the Configuration File Manager (CFM) application. CFM provides a file repository on the management server in **/cfmroot** to contain all configuration files that are shared by CFM Managed nodes.

**Note:** The **cfmupdatenode** command cannot be used for Minimally Managed nodes (Linux nodes with no CSM or RSCT code).

The **cfmupdatenode** command is run on the management server and causes immediate distribution of the configuration files located in the server file repository (**/cfmroot**). The target nodes must be supplied to the **cfmupdatenode** command using **-a**, **-N**, **--file**, **-n**, or positional arguments.

Symbolic links can be placed in the **/cfmroot** directory, but the **cfmupdatenode** command will transfer the target of the symbolic link. If symbolic links need to be maintained throughout the cluster, they should be created with *filename.post* scripts.

A file in the server file repository can either be sent to all nodes, to a defined node group, or to a specific host name. To associate a file with a node group, add a *.\_GroupName* extension to the file name. This file will then only be distributed to nodes in this CSM node group. To associate a file with a specific host name add a *.\_hostname* extension to the file – the file will only be transferred to the node corresponding to that host name. During file transfer the *.\_GroupName* or *.\_hostname* extension is removed. The *.\_GroupName* or *.\_hostname* extension must come at the end of the directory or file name, and it cannot contain the */* character.

**Note:** Adding a *.\_GroupName* extension to a file creates a persistent association between a CSM node group and the file. This association should be considered when changing node group definitions.

The default remote transfer protocol for the **cfmupdatenode** command is the protocol stored in the CSM database (see the **csconfig** command). However, this value can be overridden by the **DSH\_REMOTE\_CMD** environment variable or by specifying a remote shell after the **-r** option. CFM can use any remote shell with **rsh** syntax that allows automated remote login.

The **-q** option queries and displays any CFM files that are out of date across the cluster. The query can also compare the last time CFM updated a node to the last file modification time in **/cfmroot**. This quickly displays which nodes are out of date and can be invoked through the combination of the **-q** and **-s** options.

## cfmupdatenode

CFM allows file preprocessing and postprocessing; for example, to start and stop daemons. In cases where a *filename.pre* or *filename.post* script exists in the same directory on the management server as **file filename**, the *filename.pre* script or executable is run on the destination node before the configuration file is copied, and *filename.post* script is run after the configuration file is copied. The scripts are run under the same user as the **cfmupdatenode** command. All standard output and standard errors from these scripts is redirected by the **cfmupdatenode** command to both standard output and standard errors on the management server. Standard output is only displayed if the **cfmupdatenode** command is run with the **-v** verbose option. These scripts can be written in any shell-based scripting language.

**Note:** The *filename.pre* and *filename.post* scripts are retained on each node in the **/var/opt/csm/cfmlocal/** directory. File permissions should be set so that these are not accessed by unauthorized users. The file permissions should be set in **/cfmroot** so that they will be maintained throughout the cluster.

Files transferred by CFM retain their original modification time stamp from the management server. While the cluster does not need to have synchronized time clocks for the successful operation of CFM, the administrator should exercise caution when operating CFM in younger (**-y**) mode.

CFM is designed to facilitate the distribution of individual configuration files across a cluster. It should not be used to distribute or maintain copies of entire file systems, which could result in extremely slow performance. CFM can transfer both text and binary files. However, it should be noted that CFM provides no support for Endian byte conversion between architectures.

## Parameters

None.

## Options

- a** Distributes files to all nodes in the cluster. This option cannot be used with **-n**, **-N**, **--file**, or *node\_list*. Use this option with caution in mixed clusters. Make sure the **/cfmroot** directory is set up to work with **-a**, since the nightly cron job pushes files out to all nodes.
- b** Backs up existing configuration files on nodes as *filename.OLD*.
- c** Performs binary comparison on files and transfers them if they differ.
- d** *distfile location*  
Generates a **distfile** but does not call **Rdist**, the open source utility used for file transfer. This allows the administrator to run **Rdist** with flexible options.
- f** *filename*  
Updates only the given file name. The file name should contain the absolute path name of the file and the file must reside in the **/cfmroot** directory.
- file** *filename*  
Specifies a file that contains a list of node names. If the file name specified is **-**, then the list is read from standard input. The file can contain multiple lines and each line can have one or more node names, separated by spaces.
- h** Writes the command's usage statement to standard output.

- M** *max\_child\_processes*  
Sets the number of nodes to update concurrently. The default is 32 or the value specified by the **CSM\_FANOUT** environment variable. Depending on the number of files to be transferred, **cfmupdatenode** can take time to update a node, so this number should be set accordingly. If all files are not being transferred to the target nodes, reduce the **CSM\_FANOUT** value.
- n** *node\_list*  
Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.  
  
**Note:** You cannot use the **-n** option with the **-a** option.
- N** *nodegroups*  
Resolves one or more node groups, separated by spaces or commas, and sends the resulting nodes their configuration files.
- q**  
Sends a query to determine the CFM files that are out of date across the cluster. These files will be updated by the next normal run of the **cfmupdatenode** command. If all files are current, nothing is reported.
- r** *remote shell path*  
Specifies the path to use for remote commands. This option can point to either an **rshell** or a secure remote command executable. If not specified with options, the remote shell defaults first to the value of the **DSH\_REMOTE\_CMD** environment variable and then to the CSM *RemoteShell* attribute value (see the **csmconfig** command).
- s**  
Reports which Managed nodes are up-to-date by comparing the last CFM update time with the modification time of files in the **/cfmroot** directory. Only use this option with the **-q** option.
- t** *timeout*  
Sets the timeout period (in seconds) for waiting for responses from a remote process. The default is 900 seconds.
- v | -V**  
Specifies verbose mode.
- y**  
Specifies younger mode. Files are normally updated if their modification times and sizes disagree with those of the master copy (located in **/cfmroot**). This option causes the **cfmupdatenode** command to not update files that are younger than the master copy. This option should be used only if the administrator is confident that the time clocks on the client nodes are in sync with the time clock on the management server. A warning message is printed for files that are newer than the master copy.

## Exit Status

Exit codes 1 through 4 indicate that the **cfmupdatenode** command cancelled the file transfer when it received this error. Exit codes 10 and higher indicate that the **cfmupdatenode** command attempted to distribute as many files as possible.

- 0** All files transferred successfully.
- 1** A usage error occurred.
- 2** Could not write **distfile** at **/tmp/cfm\_distfile** on the management server.
- 3** Another instance of the **cfmupdatenode** command is currently running.

## cfmupdatenode

- 4 Could not run **lsnode** or **nodegrp** because CTRMC is down.
- 10 **rdist** returned a non-zero exit code. Some files were probably not distributed.
- 11 **clocal** reported an error on a remote node.
- 21 A host name or node group name resolution error occurred. The administrator should consider running the CTRMC probe and checking the input to the **cfmupdatenode** command.
- 22 One or more nodes experienced severe errors, indicating that they probably did not receive any configuration files. This includes unreachable **Status = 0** nodes.
- 23 There were errors concerning file names in **/cfmroot**. Some files were probably not distributed.

## Security

The command requires root access to the cluster management server.

## Examples

1. To force CFM to run on all nodes, enter:  

```
cfmupdatenode -a
```
2. To run CFM on Managed node **clsn03**, enter:  

```
cfmupdatenode clsn03
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

<b>/cfmroot</b>	The directory on the management server that contains the cluster's configuration files.
<b>/tmp/cfm_distfile</b>	Location of the file created by the <b>cfmupdatenode</b> command, containing instructions on transferring files.
<b>/var/log/csm/cfmerror.log</b>	Location of CFM error logs.
<b>/var/log/csm/cfmchange.log</b>	Location of the log, on the management server, that contains the dates when files were updated.

## Location

**/opt/csm/bin/cfmupdatenode**

## See Also

The **csmconfig** and **updatenode** commands.

## Author

IBM CSM Development - cluster@us.ibm.com



---

## chnode Command

### Purpose

**chnode** - Changes a node definition in the CSM database.

### Synopsis

```
chnode [-h] [-v | -V] {-a | -f filename | [[-n] node_list ] | -N nodegroups | -w selectstr} Attr=value [[Attr=value...]]
```

### Description

The **chnode** command changes a node definition in the CSM database by setting one or more attribute values. Attribute values can be set in the database for this node definition by specifying attribute and value pairs on the command line in the form *Attr=value*. If the value is a string that contains spaces or other special characters, the value must be enclosed in quotation marks. The *node\_list* parameter can be specified by host name or IP address. If **-w selectstr** is specified, the **chnode** command uses the string in the "where" part of an SQL select statement against the database of nodes, and changes the nodes that are matched.

The **chnode** command can be run only from the management server to change a node definition in the CSM database on the management server.

### Parameters

*Attr=value*

Specifies a CSM node attribute value. For a list of valid attributes to use in the *Attr=value* pairs, see the **nodeattributes** man page. *Attr=value* pairs must be specified last on the command line.

### Options

**-a** Changes all the nodes. Note that you cannot use the **-a** option with the **-n** option.

**-f filename**

Specifies a file that contains a list of nodes names. If the *filename* value is **-** then the list is read from standard input. The file can contain multiple lines and each line can have one or more node names separated by spaces, commas, or both.

**-h** Writes the command's usage statement to standard output.

**-n node\_list**

Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

Note that you cannot use the **-n** option with the **-a** option.

**-N nodegroups**

Changes the nodes that are in the node groups specified. The node group is evaluated by the **nodegrp** command. Multiple node groups can be specified within quotation marks and separated by commas or spaces.

**-v | -V**

Writes the command's verbose messages to standard output.

## chnode

### **-w** *selectstr*

Changes the nodes that match the "where" part of the select string. It is easiest to put the whole string in double quotation marks, especially if you need to put attribute values in single quotation marks (when they are strings). To specify all nodes use a \* instead of a *selectstr* value.

## Exit Status

- 1** A combination of options and arguments has been entered that is not valid.
- 12** Node not found.
- 13** No nodes matched the specified string.

If an error occurs in the RMC layer, the RMC return code is returned as the exit status.

## Security

The command requires root access to the cluster management server.

## Examples

To change the operating system version in the CSM database to 8.0 for the node named **websvr**, enter:

```
chnode websvr InstallDistributionVersion='8.0'
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

**/opt/csm/bin/chnode**

## See Also

The **definenode**, **lsnode**, **nodeattributes**, **nodegrp**, and **rmnode** man pages.

See the **rmcli** man page in the *IBM RSCT for Linux: Technical Reference* for information on *Attr=value* syntax.

See the *IBM RSCT for Linux: Administration Guide* for information on selection string syntax.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## chrconsolecfg Command

### Purpose

**chrconsolecfg** – Removes, adds, or rewrites console entries in the Conserver configuration file.

### Synopsis

**chrconsolecfg** [-h] [-i] [-a | -r] {[-n *node\_list*] [-N *nodegroups*]}

### Description

The **chrconsolecfg** command is a maintenance tool for updating entries in the Conserver configuration file. If the command is run with the **-a** option, the Conserver configuration file is rewritten using all nodes in the CSM database. Otherwise, specific nodes to process are specified with the **-n** or **-N** options. If the **-r** option is used, the entries for the specified nodes are removed from the configuration file; otherwise entries are added. If a node name to be added already exists in the configuration file, the existing entry will be removed and a new one will be added. If a node name to be removed does not exist in the configuration file, a message will be returned. The command will continue, but will exit with a return code of **-1**.

CSM software automatically runs the **chrconsolecfg** command to update the configuration file whenever new nodes are added to, or removed from, the cluster.

The **-i** option causes the "allowed:" tag of the configuration file to be rewritten with all of the network interfaces defined on the Management Server (excluding the loopback interface). This tag is used by Conserver to identify which network domains it accepts commands from. This option can be used by itself, or in combination with the **-a**, **-r**, **-n**, or **-N** options.

### Parameters

None.

### Options

- a**      Rewrites the Conserver configuration file using all nodes in the CSM database. Note that you cannot use the **-a** option with the **-n** option.
- h**      Displays command usage information.
- i**      Rewrites the interfaces entry of the configuration file.
- n** *node\_list*  
Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.  
  
Note that you cannot use the **-n** option with the **-a** option.
- N** *nodegroups*  
Specifies node names in the specified node groups to add or remove in the Conserver configuration file.
- r**      Removes entries from the Conserver configuration file.

## chrconsolecfg

### Exit Status

- 0 The command has run successfully.
- 1 An error has occurred with one or more of the remote console commands.
- 1 A request was made to remove a node from the configuration file, but the node was not present in the file.

### Security

The command requires root access to the cluster management server.

### Examples

1. To add selected nodes to the Conserver configuration file, overwriting any existing entries, enter:

```
chrconsolecfg -n node1,node2,node4
```

2. To add selected node groups to the Conserver configuration file, enter:

```
chrconsolecfg -N nodegroup4,nodegroup6
```

3. To regenerate the Conserver configuration file with all nodes in the CSM database, enter:

```
chrconsolecfg -a
```

4. To remove selected nodes from the Conserver configuration file, enter:

```
chrconsolecfg -r -n node2,node4
```

5. To remove selected node groups from the Conserver configuration file, enter:

```
chrconsolecfg -r -N nodegroup3,nodegroup5
```

**Note:** The **-a** and **-r** options are mutually exclusive. You cannot use **chrconsolecfg -a -r** to remove all nodes in the Conserver configuration file.

6. To rewrite the "allowed: " line of the configuration file, enter:

```
chrconsolecfg -i
```

### Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

### Files

#### **/etc/opt/conserver/conserver.cf**

Configuration file. **DO NOT EDIT THIS FILE.** This file is automatically updated by CSM as nodes are added to or removed from the cluster. Improper modifications to this file could prevent remote console access to the nodes.

### Location

**/opt/csm/bin/chrconsolecfg**

## See Also

- The **rconsole** and **rconsolerefresh** commands.
- The *IBM CSM for Linux: Hardware Control Guide*.
- The Conserver serial console application Web site (<http://www.conserver.com>).

## Author

IBM CSM Development - [cluster@us.ibm.com](mailto:cluster@us.ibm.com)

---

## chsnmp Command

### Purpose

**chsnmp** - Sets the SNMP agent configuration information for nodes on xSeries and BladeCenter servers.

### Synopsis

```
chsnmp [-h] | [-v] {-a | -n node_list | -N nodegroups} [-m name,...] [-M name,...] [-d] [-c contact] [-l location] [-i address,...] {-m name,...} [-l address,...] [-m name,...] [-e event,...] [-r event,...] {enable_agent | disable_agent | enable_traps | disable_traps}
```

### Description

The **chsnmp** command runs on the management server to set SNMP agent configuration information for the specified nodes on xSeries and BladeCenter servers.

**Note:** This command is not supported for Linux on pSeries clusters.

The command uses the CSM database to determine the nodes and their service processor information. The host name used must be the name specified when defining the nodes in the CSM database. The command prints a response for each node, specifying the node name with the return code. The **-e** and **-r event** operand values can include the following predefined event names:

```
ALL
CRITICAL_ALL
CRITICAL_TEMPERATURE
CRITICAL_VOLTAGE
CRITICAL_TAMPER
CRITICAL_MULTIPLE_FAN
CRITICAL_POWER_FAILURE
CRITICAL_HARD_DRIVE
CRITICAL_VRM_FAILURE
NONCRITICAL_ALL
NONCRITICAL_RPS
NONCRITICAL_SINGLE_FAN
NONCRITICAL_TEMPERATURE
NONCRITICAL_VOLTAGE
SYSTEM_ALL
SYSTEM_PFA_ALERT
SYSTEM_WATCHDOG_LOADER
SYSTEM_WATCHDOG_OS
SYSTEM_WATCHDOG_POST
SYSTEM_APPLICATION
SYSTEM_BOOT_FAILURE
```

Note that the **chsnmp** command can only be used on nodes for which **xseries** is the power method. The command is currently not supported for IntelliStation™ workstations.

### Options

**-a** Runs the command on all nodes on xSeries and BladeCenter servers in the cluster. Note that you cannot use the **-a** option with the **-n** option.

**-c contact**  
Sets the system contact name.

**-d** Sets the defaults for the first community *name* to *public* and the first community *address* to the address of the management server.

**disable\_agent**

Disables the SNMP agent.

**disable\_traps**

Disables trap-directed notification.

**-e event...**

Sets the event trigger.

**enable\_agent**

Enables the SNMP agent.

**enable\_traps**

Enables trap-directed notification.

**-h** Writes usage information to standard output.

**-i address...**

Sets the community addresses. You must also specify **-m**.

**-I (uppercase i) address...**

Removes the community addresses. You must also specify **-m**.

**-l (lowercase L) location**

Sets the system location.

**-m name...**

Sets the community names.

**-M name...**

Removes the community names.

**-n node\_list**

Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

Note that you cannot use the **-n** option with the **-a** option.

**-N nodegroups**

Specifies one or more node groups on which to run the command.

**-r event...**

Clears the event trigger.

**-v**

Turns verbose mode on. When you specify this option, tracing is turned on for the duration of the command. After the command is run, tracing is returned to its previous state.

## Exit Status

- 0** The command has run successfully.
- 1** An error has occurred.

## Security

The command requires root access to the cluster management server and a user ID with access to the IBM.NodeHwCtrl resource class in the RMC **ctrmc.acls** ACL file.

## chsnmp

This command may require a **systemid** file. For more information, see the **systemid** command.

## Examples

1. To enable the SNMP agent on nodes clsn01 and clsn02, enter:

```
chsnmp -n clsn01,clsn02 enable_agent
```

Output is similar to the following example:

```
clsn01 enable_agent complete rc=0  
clsn02 enable_agent complete rc=0
```

2. To set the community name for host clsn01 to Community1, enter:

```
chsnmp -n clsn01 -m Community1
```

Output is similar to the following example:

```
clsn01 set_community complete rc=0
```

3. To remove the community name Community1 for host clsn01, enter:

```
chsnmp -n clsn01 -M Community1
```

Output is similar to the following example:

```
clsn01 clear_community complete rc=0
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

Output files similar to those shown in the Examples section above.

## Location

```
/opt/csm/bin/chsnmp
```

## See Also

The **lssnmp** command.

## Author

IBM CSM Development - cluster@us.ibm.com



## copycsmpkgs Command

### Purpose

**copycsmpkgs** - Copies CSM packages and prerequisites to the `/csminstall` directory.

### Synopsis

```
copycsmpkgs [-h] [-v | -V] [-c] -p pkg_path [Attr=value ...]
```

### Description

The **copycsmpkgs** command is used when the attributes of the nodes to be installed do not match the attributes of the management server. For example, the command should be used if the CSM version to be installed on the nodes is different than the CSM version on the management server.

The **copycsmpkgs** command copies install packages from the directories specified with the `-p` option. The following install packages are copied:

<b>CSM Packages</b>	CSM product packages are installed on the nodes.
<b>Open Source Packages</b>	Packages that are not part of the operating system, but are shipped with CSM or downloaded and must be installed on the nodes.
<b>Operating System Packages</b>	Packages that are part of the base operating system and must be installed on the nodes. If any of these RPM packages are missing, you will be prompted to insert the distribution CD-ROMs.

### Parameters

The following parameters can be specified on the command line. Any omitted parameters default to the value for the management server.

*Attr=value ...*

Specifies one or more *Attr=value* pairs separated by spaces. *Attr=value* pairs must be specified last on the command line. See the **nodeattributes** man page for the values that should be used for each attribute. The valid attributes are:

*InstallCSMVersion*

The CSM version.

*InstallOSName*

The operating system. This attribute is required.

*InstallDistributionName*

The operating system distribution. This attribute is required.

*InstallDistributionVersion*

The operating system version. This attribute is required.

*InstallPkgArchitecture*

The operating system package architecture. This attribute is required.

### Options

`-c` Forces the command to copy all required files from the *pkg\_path*. Normally,

## copycsmpkgs

**copycsmpkgs** only copies files that are newer or the same as those in **/csminstall**, or files that do not exist in **/csminstall**. The **-c** option forces **copycsmpkgs** to copy all files regardless of their version.

- h** Writes usage information to standard output.
- p *pkg\_path***  
Specifies one or more colon-separated directories containing CSM and RPM packages.
- v | -V**  
Writes verbose messages to standard output.

## Security

The command requires root access to the cluster management server.

## Examples

To install CSM on Linux nodes that are running Red Hat 7.3, with a Linux management server running Red Hat 8.0, enter:

```
copycsmpkgs -p /images/csm:/images/rh73disk1:/images/rh73disk2 InstallOSVersion=7.3
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

**/opt/csm/bin/copycsmpkgs**

## See Also

- The **updatenode** man page.
- The **nodeattributes** man page.
- The *IBM CSM for Linux: Planning and Installation Guide*.
- The **csm.core** and **csm.server** packages.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## csm

### Purpose

**csm** - Lists the CSM commands and files. **The csm man page does not represent a command. It is provided as an informational man page for your reference only.** Each command or file listed has its own man page. Enter **man *command\_or\_file*** for detailed information on the command or file.

CSM provides the following commands:

#### **addlcsext**

Adds IBM Director Linux Cluster Support extension components.

#### **cfmupdatenode**

Distributes configuration files across a CSM cluster.

#### **chnode**

Changes a node definition in the CSM database.

#### **chrconsolecfg**

Removes, adds, or rewrites console entries in the Conserver configuration file.

#### **chsnmp**

Sets the SNMP agent configuration information for nodes on xSeries and BladeCenter servers.

#### **copycsmpkgs**

Copies CSM packages and prerequisites to the **/csminstall** directory.

#### **csmbackup**

Extracts and stores vital CSM data.

#### **csmconfig**

Displays or changes CSM cluster attributes.

#### **csmrestore**

Restores CSM data backed up by the **csmbackup** command.

#### **csmsetupks**

Configures Kickstart and defines the nodes to be installed in the CSM database.

#### **csmsetupyast**

Configures AutoYaST and defines the SuSE and SuSE SLES nodes to be installed in the CSM database.

#### **csmsis2yast**

Migrates SIS configuration to AutoYaST.

#### **csmstat**

Provides a snapshot of cluster node reachability, power status, and network interface status.

#### **csm2nimgrps**

Creates Network Installation Manager (NIM) machine group definitions corresponding to CSM cluster node group definitions.

**dcem** Opens the DCEM GUI for running a command or script on multiple distributed nodes.

#### **definenode**

Defines the nodes in a CSM cluster.

- dsh** Concurrently issues remote shell commands to multiple hosts and formats results.
- dshbak**  
Presents formatted output from the **dsh** command.
- getadapters**  
Collects information on the network adapters installed on a node.
- installms**  
Installs the Cluster Systems Management (CSM) management server software.
- installnode**  
Installs the Linux operating system and CSM on nodes.
- lshwinfo**  
Collects node information from one or more hardware control points.
- lshwstat**  
Collects environmental and Vital Product Data (VPD) information from nodes on xSeries hardware.
- lsnode**  
Lists the node definitions in the CSM database.
- lssnmp**  
Collects SNMP agent configuration information from nodes on xSeries and BladeCenter servers.
- mgmtsvr**  
Displays or sets the management server entry for the Managed node.
- monitorinstall**  
Displays CSM node installation status.
- nodegrp**  
Manages node group definitions in the Cluster Systems Management (CSM) database.
- predefined-condresp**  
Defines default conditions and responses for monitoring.
- predefined-nodegroups**  
Defines predefined node groups to their initial states.
- probemgr**  
Runs CSM cluster system diagnostics.
- rconsole**  
Opens a console window to a remote node.
- rconsolerefresh**  
Refreshes the Conserver daemon.
- reventlog**  
Collects service processor log information from nodes on xSeries hardware.
- rmlcsext**  
Removes IBM Director Linux Cluster Support extension components.
- rmnode**  
Removes node definitions from the CSM database.
- rpower**  
Controls power for remote nodes.

**smsupdatenode**

Updates open source RPM packages on Linux nodes.

**systemid**

Stores the user ID and password required for internal programs to access hardware.

**uninstallms**

Uninstalls the CSM management server software.

**updateisvr**

Updates CSM installation files across multiple install servers.

**updatenode**

Installs CSM, adds the node to the cluster, runs Linux software maintenance (SMS), and distributes CFM files.

The following files are provided with CSM:

**kscfg.tmpl File**

Linux operating system configuration file template used by the **csmsetupks** command.

**nodedef**

The CSM node definition file.

The following informational man pages are provided with CSM:

**csm** Lists the CSM man pages for commands and files.

**nodeattributes**

Displays the node attributes in the CSM database.

**nodorange**

Displays the short hand for specifying list of nodes to CSM commands.

---

## csmbackup Command

### Purpose

**csmbackup** - Copies and stores vital CSM data.

### Synopsis

**csmbackup** [-h] [-v | -V] [-d *directory*] [-f *filename*] [-r *resource*]

### Description

The **csmbackup** command copies vital CSM data from the management server and stores the data in the directory specified by **-d** option. The command backs up a CSM management server in case of a hardware problem, or if the management server is being changed to another machine.

**Note:** This command is not supported for Linux on pSeries clusters.

The **csmbackup** command nearly replicates the CSM state of the management server. The command backs up the nodes, node groups, conditions and responses, and condition and response associations. Customization scripts and DCEM scripts are also copied.

The command output includes a list of additional tasks that need to be performed manually to complete the backup. The command does not do some tasks because certain data should be copied to a backup storage device manually using commands such as **scp**, **rsync**, or **rcp**, or other methods of replicating data. However, a list of additional files for the **csmbackup** command to store can be specified with the **-f** option.

### Options

**-d** *directory*

Specifies the directory where CSM backup data will be stored. If the **-d** option is not specified, the data is stored in **/var/opt/csm/csmdata**.

**-f** *filename*

Specifies a file that contains a list of additional CSM files to back up.

**-h** Displays command usage information.

**-r** *resource*

If the **-r** option is used, only the specified resource is backed up. This option is useful for registering ERRM event responses to only back up a specified resource class when a resource class changes. The *resource* value can be **Sensor**, **EventResponse**, **Condition**, **Association**, **ManagedNode**, **NodeGroup**, or **DmsCtrl**. Only one resource can be specified at a time.

The **csmbackup** command will only back up a specific CSM resource class or file if you use the **-r** option followed by one resource class or a file containing a list of files; using **-f** for example. The following arguments can be used:

**Sensor**

Backs up the information found in the IBM.Sensor resource class.

**EventResponse**

Backs up event responses, which include the responses returned by the **lsresponse** command and scripts used by the ERRM responses.

**Condition**

Backs up information returned by the **lsresponse** command.

**Association**

Backs up information regarding the mapped conditions and responses that are currently configured.

**ManagedNode**

Backs up the data returned by the **lsnode** command.

**NodeGroup**

Backs up node group information.

**DmsCtrl**

Backs up information that is stored by the **csmconfig** command.

*file* If the **-r** option is followed by a file that exists then the **csmbackup** command will back up the contents of that file.

**-v | -V**

Writes verbose messages to standard output.

**Exit Status**

- 0** The command completed successfully.
- 1** An error has occurred.

**Security**

The command requires root access to the cluster management server.

**Examples**

1. To back up all node groups to **nodegrp.csmbak**, enter:  

```
csmbackup -r NodeGroup
```
2. To back up CSM data to **/tmp/csmdata** and copy the files in **/myfile** to **/tmp/csmdata/usrbak/etc**, enter:  

```
csmbackup -d /tmp/csmdata -f /tmp/myFile
```

**Implementation Specifics**

This command is part of IBM Cluster Systems Management (CSM) software.

**Files**

**/var/log/csm/csmbackup.log**

Location of the log output from the **csmbackup** command.

**Location**

**/opt/csm/bin/csmbackup**

**See Also**

The **csmrestore** command.

**csmbackup**

| **Author**

|

IBM CSM Development - cluster@us.ibm.com



---

## csmconfig Command

### Purpose

**csmconfig** - Displays or changes certain cluster attributes that affect CSM.

### Synopsis

**csmconfig** [-v] -V

**csmconfig** [-v] -d *attribute*

**csmconfig** [-v] -L [*license-key-file*]

**csmconfig** [-h] [-c] [-u **yes** | **no**] [-m *cluster-type-model*] [-p *PowerMethodMode*] [-s *cluster-serialnum*] [-r *remote-shell-path*] [-R **yes** | **no**] [-v] [*Attr=value ...*]

### Description

The **csmconfig** command, which is run on the management server, allows you to display or list the cluster attributes that CSM uses. One or more of the cluster attributes can be set using the options that follow. If no option is specified, the value of all cluster attributes is displayed, including the expiration date of the license key, if applicable.

### Parameters

*Attr=value...*

Specifies cluster attributes that should be changed in the **IBM.DmsCtrl** RMC resource class. This is an alternative to using a specific option, or can be used for a cluster attribute for which there is no command line option. Each *Attr=value* pair should have no spaces, or the parts containing spaces should be enclosed in quotation marks. A null value can be specified by *Attr=*. *Attr=value* pairs must be specified last on the command line. The CSM attributes that can be specified with *Attr=value* are:

#### **AddUnrecognizedNodes**

Specifies if the DMS resource manager should accept unexpected requests from systems, to manage them and allow them to join the CSM cluster. The default is **no**. This cluster attribute can also be set using the **-u** option.

#### **ClusterSNum**

Sets the cluster serial number to this string. This value is used for service. This cluster attribute can also be set using the **-s** option.

#### **ClusterTM**

Sets the cluster type and model to this string. This value is used for service, and should be in the form #####-###. This cluster attribute can also be set using the **-m** option.

#### **ExpDate**

Specifies the expiration date for the current *try-and-buy* license key. This cluster attribute cannot be set. Use the **-L** option to install a full license key.

#### **HeartbeatFrequency**

Number of seconds between heartbeat messages sent to a node.

## csminstall

- | **HeartbeatSensitivity**  
| Number of missed heartbeat messages sent to a node to declare  
| that node unreachable.
- | **MaxNumNodesInDomain**  
| Specifies the maximum number of nodes allowed in this CSM  
| cluster. This is affected by the license key that is installed using the  
| **-L** option.
- | **PowerMethodMode**  
| Sets the mode for retrieving node power status. Valid values are  
| **mixed**, **polling**, **events**, and **none**.
- | **RegSyncDelay**  
| Delay in seconds between the time the cluster data is updated in  
| memory and written onto the disk.
- | **RemoteShell**  
| Specifies the path of the remote shell command that CSM should  
| use to run commands on the nodes. The default is **/usr/bin/ssh**.  
| This cluster attribute can also be set using the **-r** option.
- | **SetupRemoteShell**  
| Indicates whether CSM should try to automatically set up security  
| for the remote shell specified in the RemoteShell cluster attribute.  
| The default is **yes**. This cluster attribute can also be set using the  
| **-R** option.

The following list shows the command line options and their corresponding cluster attribute names:

- u** *AddUnrecognizedNodes*
- s** *ClusterSNum*
- m** *ClusterTM*
- | **-p** *PowerMethodMode*
- r** *RemoteShell*
- R** *SetupRemoteShell*

## Options

- | **-c** Creates the **/csminstall** subdirectories and copies the required file sets to  
| the correct **/csminstall** subdirectories.
- | **-d attribute**  
| Displays just this one cluster attribute value. If the **csminstall** command is  
| invoked with no arguments, all the cluster attribute values are displayed.  
| This option should not be used with any others except **-v**.
- | **-h** Displays usage information to standard output.
- | **-L license-key-file**  
| Provide a new license key file (ECF) to CSM. If *license-key-file* is omitted,  
| the default Try-and-Buy license key will be used. The current expiration date  
| can be viewed by running the **csminstall -d ExpDate** command. This  
| option should not be used with any others except **-v**.
- | **-m cluster-type-model**  
| Sets the cluster type and model to this string. This value is used for service.

- p** *PowerMethodMode*  
Sets the mode for retrieving node power status. Valid values are **mixed**, **polling**, **events**, and **none**.
- r** *remote-shell-path*  
Specifies the path of the remote shell command that CSM should use to run commands on the nodes. The default is **/usr/bin/ssh**.
- R** **yes | no**  
Indicates whether CSM should try to automatically set up the security for the remote shell specified in the RemoteShell parameter. The default is **yes**.
- s** *cluster-serialnum*  
Sets the cluster serial number to this string. This value is used for service.
- u** **yes | no**  
Indicates whether the DMS resource manager should accept unexpected requests from systems, to manage them and bring them into the CSM cluster. The default is **no**.
- v**  
Writes the command's verbose messages to standard output.
- V**  
Displays the current CSM version that is installed on the management server. This option should not be used with any others except **-v**.

## Exit Status

- 0** The command has run successfully.
- 1** The combination of options is not valid.
- 2** Did not install the specified *try-and-buy* license key because a license key is already installed.
- 3** Did not install the specified license key because the License Agreement was not accepted.

## Security

The command requires root access to the cluster management server.

## Examples

1. To show the values currently set, enter:  
`csmconfig`
2. To change the default remote shell to rsh, enter:  
`csmconfig -r /usr/bin/rsh`
3. To accept the *try-and-buy* license:  
`csmconfig -L`

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

**/opt/csm/bin/csmconfig**

## Author

IBM CSM Development - cluster@us.ibm.com

---

## csmrestore Command

### Purpose

**csmrestore** - Restores CSM data backed up by the **csmbackup** command.

### Synopsis

**csmrestore** [-h] [-v | -V] [-d *directory*]

### Description

The **csmrestore** command restores the files that were copied by the **csmbackup** command into the specified directory or the default directory **/var/opt/csm/csmdata**. The CSM management server must be completely installed and configured by the administrator in advance; the **csmrestore** command does not install any CSM code on the management server. The **csmrestore** command only restores the nodes, node groups, conditions and responses, and condition and response associations on the management server. Customization scripts and DCEM scripts are also restored.

**Note:** This command is not supported for Linux on pSeries clusters.

### Options

**-d** *directory*

Specifies the directory from which to restore CSM backup data. Data in this directory should have been placed there by the **csmbackup** command.

**-h** Displays command usage information.

**-v | -V**

Writes verbose messages to standard output.

### Exit Status

**0** The command completed successfully.

**1** An error has occurred.

### Security

The command requires root access to the cluster management server.

### Examples

1. To restore information from the default backup directory **/var/opt/csm/csmdata**, enter:

```
csmrestore
```

2. To restore data from any other directory, use the **-f** option:

```
csmrestore -f /mnt/backupfiles/csm/
```

### Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

### Location

**/opt/csm/bin/csmrestore**

| **See Also**

| The **csmbackup** command.

| **Author**

| IBM CSM Development - cluster@us.ibm.com

---

## csmsetupks Command

### Purpose

**csmsetupks** – Configures Kickstart and defines the nodes to be installed in the CSM database.

### Synopsis

**csmsetupks** [-h]

**csmsetupks** [-p *[path]* | -x] [-k *kickstart file*] {-P | -a | [-N *nodegroups*] [-n *node\_lisf*]} [-v | -V] [*Attr=value ...*]

### Description

The **csmsetupks** command collects configuration information and configures Kickstart to install the Red Hat Linux operating system and Cluster Systems Management (CSM). After running the **csmsetupks** command, the system administrator must run the **installnode** command to install the nodes in the cluster.

**Note:** This command is not supported for Linux on pSeries clusters.

The **csmsetupks** command copies the contents of the distribution CD-ROMs or the contents from each path specified by the **-p** option. The contents of the disks or directories are copied to

*/csminstall/InstallOSName/InstallDistributionName/InstallDistributionVersion/InstallPkgArchitecture*. For example,  
*/csminstall/Linux/RedHat/8.0/i386*

All RPM packages that are copied will be located in the **/RPMS** subdirectory; for example, **/csminstall/Linux/RedHat/8.0/i386/RPMS**. If a path is specified with the **-p** option, the path should point to directories that are equivalent to the root directory of a Red Hat distribution CD-ROM. For example, the **RedHat/RPMS** directory should exist inside the specified directories.

The **csmsetupks** command can be used to set up Kickstart for different versions of the Red Hat Linux operating system. The nodes specified using the **-a**, **-P**, **-N** or **-n** options and the CD-ROM or path specified dictate which version of the RPM files are copied. The **csmsetupks** command should be run only in a given instance for nodes with a common Red Hat version. The Red Hat version information is stored in the *InstallDistributionName*, *InstallDistributionVersion* and *InstallPkgArchitecture* parameters in the **ManagedNode** class. Run **lsnode -l** to list the attributes.

If the **-x** option is used, the Red Hat distribution CDs are not copied. The **-x** option should be used only if you have already run the **csmsetupks** command and the Red Hat distribution CDs have already been copied.

The **csmsetupks** command uses a template to generate the Kickstart configuration file. The *InstallTemplate* attribute in the **ManagedNode** class identifies the template file to be used for a node. If this attribute is not set, **csmsetupks** will use a default template file determined by which operating system level is being installed. The default template is located in:

*/opt/csm/install/kscfg.tmpl.InstallDistributionNameInstallDistributionVersion*. For example, the Red Hat 8.0 template file is named

`/opt/csm/install/kscfg.tmpl.RedHat8.0`. Template files can be modified to suit your installation; see the `kscfg.tmpl` file and the *IBM CSM for Linux: Planning and Installation Guide*.

The **csmsetupks** command also allows the user to specify which Kickstart configuration template file it should use for all nodes specified in that invocation of the command using the `-k` option. You can also indicate the kickstart template file to use by setting the *InstallTemplate* attribute for each node to the path name of the kickstart template file. Set this attribute before running **csmsetupks**. To create a template file, copy the default template and customize it as needed. The sections marked

```
### DO NOT ERASE THIS SECTION ###
```

must remain in the template file.

The **csmsetupks** command will update the *InstallTemplate* attribute in the **ManagedNode** class with the path and file name of the template file used to generate the Kickstart configuration file for that node.

The generated Kickstart configuration file is located in `/csminstall/csm/InstallCSMVersion/kickstart.InstallDistributionName\InstallDistributionVersion/NodeIPAddr-kickstart`. For example:  
`/csminstall/csm/1.2/kickstart.Linux8.0/9.117.241.32-kickstart`

The **csmsetupks** command performs the following tasks:

- Copies the contents of the Red Hat CD-ROMs or directories specified by `-p`, and prompts for CD-ROMs as needed. The contents of the CD-ROMs or directories include RPM packages and data files. If the `-x` option is provided, nothing is copied. The contents of the CD-ROMs or directories are copied into `/csminstall/InstallOSName/InstallDistributionName/InstallDistributionVersion/InstallPkgArchitecture`. For example,  
`/csminstall/Linux/RedHat/8.0/i386`
- Starts the DHCP, NFS and TFTP daemons, if they are not already running.
- Sets up the `/etc/dhcpd.conf` file for MAC address collection and full installation of the nodes. The first time **csmsetupks** runs, it replaces any existing `/etc/dhcpd.conf` file. From then on, the **csmsetupks** command just updates the existing file. The original file is saved to `/etc/dhcpd.conf.precsm`. After the **csmsetupks** command runs the first time, you can customize the `/etc/dhcpd.conf` file provided you do not alter the sections marked with:  

```
# CSM RANGE (DO NOT REMOVE THIS LINE)

### CSM STATIC ENTRIES (DO NOT ERASE THIS LINE) ###
```
- Creates the files that are necessary for network boot in `/csminstall`.
- Creates a Kickstart configuration file for each node in `/csminstall`.
- Creates CSM configuration files and scripts in `/csminstall`.
- Gets the MAC addresses of all specified nodes by running the **getadapters** command. The **getadapters** command will first attempt to get the MAC addresses from all **Managed** and **MinManaged** nodes using the **dsh** command. If no **dsh** communication is available, or if the nodes are not in **Managed** or **MinManaged** state, **getadapters** will collect the MAC addresses using the default *MAC\_Method*, which can take time to run because some methods will reboot all the nodes in parallel. If it has a problem collecting MAC addresses, it times out after approximately five minutes.
- Sets the *InstallMethod* attribute to *kickstart* for each node provided.

## csmsetupks

- To automatically obtain MAC addresses from x445 servers using **getadapters** or **csmsetupks**, the management server must be running the 2.4.9-e.24smp or 2.4.9-e.24summit kernel.

Attribute values can be set by specifying attribute value pairs on the command line in the form *Attr=value*. If the value is a string that contains spaces or other special characters, the value must be enclosed in quotation marks. Valid attributes are:

- *Gateway*
- *Nameservers* (a comma-separated list of nameserver host names or IP addresses)

Unless otherwise noted, the default value for the attributes is the value the attributes have on the management server. These values are used when creating the */etc/dhcpd.conf* and *node-ipaddr-kickstart* files.

## Parameters

*Attr=value ...*

Specifies one or more *Attr=value* pairs, separated by spaces. *Attr=value* pairs must be specified last on the command line. Valid attributes are:

*Gateway* Specifies a global default gateway value to use in the */etc/dhcpd.conf* file for installation. If there are additional gateways (routers) for different subnets that individual nodes are connected to, and these gateways are configured on the management server, the individual gateways will be determined automatically by CSM.

*Nameservers* Specifies the nameservers to use in the *dhcpd.conf* file and in the Kickstart configuration file.

## Options

**-a** Sets up Kickstart for all nodes. This option cannot be used with the **-n**, **-N** or **-P** options.

**-h** Writes usage information to standard output.

**-k** *kickstart file*

Specifies a Kickstart configuration template file. The default is */opt/csm/install/kscfg.tmpl.RedHatversion*. To create a template file, copy the default template and customize it. The sections marked

```
### DO NOT ERASE THIS SECTION ###
```

must remain in the template file.

**-n** *node\_list* Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

Note that you cannot use the **-n** option with the **-a** or **-P** option.

**-N** *nodegroups*

Specifies a comma-separated list of node groups for Kickstart setup. This option cannot be used with the **-a** or **-P** options.



- p** *path...* Specifies one or more directories, separated by colons, that contain copies of the Red Hat CD-ROMs (*path1:path2:path3*). The default is **/mnt/cdrom**.
- P** Sets up Kickstart for all nodes with *Mode* attributes set to **PreManaged**.
- v | -V** Writes verbose messages of the command to standard output.
- x** Specifies to not copy Red Hat disks. This option cannot be used with the **-p** option.

## Security

The command requires root access to the cluster management server.

## Examples

1. To set up Kickstart for all PreManaged nodes, enter:  

```
csmsetupks -P
```
2. To set up Kickstart for all nodes in the *mynodes* node group and node *clsn02*, enter:  

```
csmsetupks -N mynodes -n clsn02
```
3. To set up Kickstart for a single node and to find RPM Packages in a list of directories, enter:  

```
csmsetupks -n clsn02 -p /images/disk1:/images/disk2
```
4. To set up Kickstart for a single node, but not copy any Red Hat files, enter:  

```
csmsetupks -n clsn02 -x
```
5. To set up Kickstart for all nodes in a node group named **computeNodes** and give it a customized kickstart file, enter:  

```
csmsetupks -N computeNodes -k /tmp/computeNodeKickStartFile
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

### **/var/log/csm/csmsetupks.log**

Location of the log file on the management server for the **csmsetupks** command. Up to five copies of this log are maintained. Old log files receive a numeric suffix up to 4.

### **/opt/csm/install/kscfg.tmpl.InstallDistributionNameInstallDistributionVersion**

Location of the Kickstart configuration file template.

### **/etc/dhcpd.conf**

Location of the DHCP configuration file.

### **/etc/dhcpd.conf.precsm**

Location of the DHCP configuration file that is backed up the first time the **csmsetupks** command is run.

## csmsetupks

### **/etc/dhcpd.conf.bak**

This back up file is created each time the **csmsetupks** command modifies the **/etc/dhcpd.conf** file.

### **/tftpboot/pxelinux.cfg/node-ipaddr-in-hex**

Location of the **PXE** configuration file, where *node-ipaddr-in-hex* represents the node's IP address in hexadecimal format (the **/usr/bin/gethostip** tool translates IP addresses to hexadecimal format). This file is modified by the **csmsetupks** and **installnode** commands during different phases of the installation.

### **/csminstall/csm/InstallCSMVersion/kickstart.\**

### **InstallDistributionNameInstallDistributionVersion/node-ipaddr-kickstart**

Location of the Kickstart configuration file generated for the node.

## Location

**/opt/bin/csmsetupks**

## See Also

- The **definnode**, **getadapters**, **installms**, and **installnode** commands.
- The **kscfg.tmpl.InstallDistributionNameInstallDistributionVersion** file.
- The *IBM CSM for Linux: Planning and Installation Guide* for more information on the installation processes.
- The *IBM CSM for Linux: Hardware Control Guide* for information on hardware control.

## Author

IBM CSM Development - cluster@us.ibm.com

## csmsetupyast Command

### Purpose

**csmsetupyast** – Configures AutoYaST and defines the SuSE and SuSE SLES nodes to be installed in the CSM database.

### Synopsis

**csmsetupyast** [-h]

**csmsetupyast** [-p *[path]* | -x] [-k *autoyastcfg\_file*] {-P | -a | [-N *nodegroups*] [-n *node\_lisf*]} [-v | -V] [*Attr=value ...*]

### Description

The **csmsetupyast** command collects configuration information and configures AutoYaST to install the SuSE or SuSE SLES Linux operating system and Cluster Systems Management (CSM). After running the **csmsetupyast** command, you must run the **installnode** command to install the nodes in the cluster.

The **csmsetupyast** command copies the contents of the distribution CD-ROMs or the contents from each path specified by the **-p** option. The contents of the disks or directories are copied to

*/csminstall/InstallOSName/InstallDistributionName/InstallDistributionVersion/InstallPkgArchitecture*. For example,  
*/csminstall/Linux/SuSE/8.1/i386*

All RPM packages that are copied will be located in the **/RPMS** subdirectory; for example, */csminstall/Linux/SuSE/8.1/i386/RPMS*. If a path is specified with the **-p** option, the path should point to directories that are equivalent to the root directory of a SuSE or SuSE SLES distribution CD-ROM.

The **csmsetupyast** command can be used to set up AutoYaST for different versions of the SuSE or SuSE SLES Linux operating system. The nodes specified using the **-a**, **-P**, **-N** or **-n** options and the CD-ROM or path specified dictate which version of the RPM files are copied. The **csmsetupyast** command should be run only in a given instance for nodes with a common version. The SuSE and SuSE SLES version information is stored in the *InstallDistributionName*, *InstallDistributionVersion* and *InstallPkgArchitecture* parameters in the **ManagedNode** class. Run **lsnode -l** to list the attributes.

If the **-x** option is used, the SuSE or SuSE SLES distribution CDs are not copied. The **-x** option should be used only if you have already run the **csmsetupyast** command and the SuSE or SuSE SLES distribution CDs have already been copied.

The **csmsetupyast** command uses a template to generate the AutoYaST configuration file. The *InstallTemplate* attribute in the **ManagedNode** class identifies the template file to be used for a node. If this attribute is not set, **csmsetupyast** will use a default template file determined by which operating system level is being installed. The default template is located in:

*/opt/csm/install/yastcfg.InstallDistributionNameInstallDistributionVersion-InstallArchitecture.xml*. For example, the SuSE 8.1 template file is named */opt/csm/install/yastcfg.SuSE8.1-i386.xml*. To generate the AutoYaST configuration file for Managed nodes, the **csmsetupyast** command uses */opt/csm/install/autoyast.chrootscript.tmpl.i386* and

## csmsetupyast

`/opt/csm/install/autoyast.postscript.tmpl.i386` on xSeries servers, or  
`/opt/csm/install/autoyast.postscript.tmpl.ppc64` on pSeries servers.

The **csmsetupyast** command also allows the user to specify which AutoYaST configuration template file it should use for all nodes specified in that invocation of the command using the **-k** option. You can also indicate the AutoYaST template file to use by setting the *InstallTemplate* attribute for each node to the path name of the AutoYaST template file. Set this attribute before running **csmsetupyast**. To create a template file, copy the default template and customize it as needed, or you can use your own AutoYaST configuration file as a template.

The **csmsetupyast** command will update the *InstallTemplate* attribute in the **ManagedNode** class with the path and file name of the template file used to generate the AutoYaST configuration file for that node.

The generated AutoYaST configuration file is located in  
`/csminstall/csm/InstallCSMVersion/autoyast.\`  
`InstallDistributionNameInstallDistributionVersion/NodeIPAddr-autoyast.xml`. For example:

`/csminstall/csm/1.3.2/autoyast.SuSE8.1/9.111.123.45-autoyast.xml`

## Parameters

*Attr=value ...*

Specifies one or more *Attr=value* pairs, separated by spaces. *Attr=value* pairs must be specified last on the command line. Valid attributes are:

*Gateway* Specifies a global default gateway value to use in the `/etc/dhcpd.conf` file for installation. If there are additional gateways (routers) for different subnets that individual nodes are connected to, and these gateways are configured on the management server, the individual gateways will be determined automatically by CSM.

*Nameservers* Specifies the nameservers to use in the `dhcpd.conf` file and in the AutoYaST configuration file.

## Options

**-a** Sets up AutoYaST for all nodes. This option cannot be used with the **-n**, **-N** or **-P** options.

**-h** Writes usage information to standard output.

**-k yastcfg\_file** Specifies an AutoYaST configuration template file, which must be an XML file. The default is  
`/opt/csm/install/yastcfg.InstallDistributionNameInstallDistributionVersion-\`  
`InstallPkgArchitecture.xml`.

**-n node\_list** Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

Note that you cannot use the **-n** option with the **-a** or **-P** option.

**-N nodegroups**

Specifies a comma-separated list of node groups for AutoYaST configuration. This option cannot be used with the **-a** or **-P** options.

- | **-p** *path...* Specifies one or more directories, separated by colons, that contain  
| copies of the SuSE or SuSE SLES CD-ROMs (*path1:path2:path3*).  
| The default is **/media/cdrom**.
- | **-P** Sets up AutoYaST for all nodes with *Mode* attributes set to  
| **PreManaged**.
- | **-v** | **-V** Writes verbose messages of the command to standard output.
- | **-x** Specifies to not copy SuSE or SuSE SLES disks. This option  
| cannot be used with the **-p** option.

## | Security

| The command requires root access to the cluster management server.

## | Examples

- | 1. To set up AutoYaST for all PreManaged nodes, enter:  
| `csmsetupyast -P`
- | 2. To set up AutoYaST for all nodes in the *mynodes* node group and node *clsn02*,  
| enter:  
| `csmsetupyast -N mynodes -n clsn02`
- | 3. To set up AutoYaST for a single node and to find RPM packages in a list of  
| directories, enter:  
| `csmsetupyast -n clsn04 -p /images/disk1:/images/disk2`
- | 4. To set up AutoYaST for a single node, but not copy any SuSE or SuSE SLES  
| files, enter:  
| `csmsetupyast -n clsn02 -x`
- | 5. To set up AutoYaST for all nodes in a node group named **computeNodes** and  
| give it a customized AutoYaST file, enter:  
| `csmsetupyast -N computeNodes -k /tmp/computeNodeAutoYastFile`

## | Implementation Specifics

| This command is part of IBM Cluster Systems Management (CSM) software.

## | Files

### | **/var/log/csm/csmsetupyast.log**

| Location of the log file on the management server for the **csmsetupyast**  
| command. Up to five copies of this log are maintained. Back level log files  
| receive a numeric suffix up to 4.

### | **/opt/csm/install/yastcfg.InstallDistributionNameInstallDistributionVersion- InstallPkgArchitecture.xml**

| Location of the AutoYaST configuration file template.

### | **/etc/dhcpd.conf**

| Location of the DHCP configuration file.

### | **/etc/dhcpd.conf.precsm**

| Location of the DHCP configuration file that is backed up the first time the  
| **csmsetupyast** command is run.

## csmsetupyast

### **/etc/dhcpd.conf.bak**

This backup file is created each time the **csmsetupyast** command modifies the **/etc/dhcpd.conf** file.

### **/tftpboot/pxelinux.cfg/node-ipaddr-in-hex**

Location of the **PXE** configuration file, where *node-ipaddr-in-hex* represents the node's IP address in hexadecimal format (the **/usr/bin/gethostip** tool translates IP addresses to hexadecimal format). This file is modified by the **csmsetupyast** and **installnode** commands during different phases of the installation.

### **/csminstall/csm/InstallCSMVersion/autoyast.\**

### **InstallDistributionNameInstallDistributionVersion/node-ipaddr\–autoyast.xml**

Location of the AutoYaST configuration file generated for the node.

## Location

**/opt/csm/bin/csmsetupyast**

## See Also

- The **csmsis2yast** command.
- The **definenode**, **getadapters**, **installms**, and **installnode** commands.
- *IBM CSM for Linux: Planning and Installation Guide* for information on installation.
- *IBM CSM for Linux: Hardware Control Guide* for information on hardware control.

## Author

IBM CSM Development - cluster@us.ibm.com

## csmsis2yast Command

### Purpose

**csmsis2yast** – Migrates SIS configuration to AutoYaST.

### Synopsis

**csmsis2yast** [-h]

**csmsis2yast** -s *sysconfig\_file* -y *autoyastcfg\_file* [-o *output\_file*] [-v | -V]

### Description

The **csmsis2yast** command converts a SIS configuration file to an AutoYaST configuration file. SIS configuration files are converted into XML code used by AutoYaST and merged into the AutoYaST configuration file, overwriting any existing disk partitioning, package, and other configuration selections in the AutoYaST file. The output is sent to standard output, or to the *output\_file* specified by the -o option.

#### Notes:

1. The **csmsis2yast** command does not convert any MACRO or NETWORK\_TYPE in a SIS configuration file.
2. This command is not supported for Linux on pSeries clusters.

### Parameters

None.

### Options

- h Writes usage information to standard output.
- o Specifies the path name to the new AutoYaST file. If this option is not used, output goes to standard out.
- s *sysconfig\_file* Specifies the path name to a SIS configuration file that includes links to a SIS partition table and package list.
- v | -V Writes verbose messages of the command to standard output.
- y *autoyastcfg\_file* Specifies the path name to an existing AutoYaST configuration file.

### Security

The command requires root access to the cluster management server.

### Examples

1. To convert a SIS configuration file to AutoYaST, enter:  

```
csmsis2yast -s /tmp/sisconfig -y /tmp/yastconfig -o /tmp/myyast.xml
```

### Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## csmsis2yast

### | Files

| **`/opt/csm/install/yastcfg.default.InstallDistributionNameInstallDistributionVersion-  
InstallArchitecture.xml`**  
| Location of the default AutoYaST configuration file.

### | Location

| **`/opt/csm/bin/csmsis2yast`**

### | See Also

| 

- The **`csmsetupyast`** command.

### | Author

| IBM CSM Development - cluster@us.ibm.com



## csmstat Command

### Purpose

**csmstat** – Provides a snapshot of cluster node reachability, power status, and network interface status.

### Synopsis

**csmstat** [-h]

**csmstat** [-l] [-a] [-S] [-s *select\_string*] [-d *delimiter*] [-n *node\_list*] [-N *nodegroups*]

### Description

The **csmstat** command gathers node reachability, power status and network interface status for one or more nodes and displays the output. The default ordering for output is by host name. If there are multiple hardware control points for a node, multiple HMCs for example, then the first hardware control point in the list is shown.

#### Notes:

1. This command does not currently support nodes on IntelliStation workstations.
2. This command is not supported for Linux on pSeries clusters.
3. Displayed LCD values are limited to 16 characters. The **rpower -l** (lowercase L) command displays the complete values.

### Parameters

None.

### Options

- a Displays attribute information for all nodes. This is the default.
- d Specifies delimiter-formatted output using the specified delimiter – colons, for example. Use this option to specify a delimiter of one or more characters. This option cannot be used with the **-a** option.
- h Displays command usage.
- l Returns LCD values for HMC-attached pSeries servers. This option cannot be used with the **-d** option.
- n *node\_list* Specifies a comma or space-separated list of node names to display attribute information. Space-separated node names must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.
- N *nodegroups* Specifies a comma or space-separated list of node groups to display attribute information. Space-separated node groups must be inside double quotes.
- s Specifies, by column headers, which columns to display. **Hostname** is displayed by default. Other values include **HWControlPoint**, **LCDS**, **Network-Interfaces**, **Status**, **PowerStatus** and **all**. This option cannot be used with the **-l** option.
- S Sorts output first by hardware control point and then by host name.

## Exit Status

### Hostname

Host name for management of the node. This value will be truncated to seventeen characters. The seventeenth character is a ~ to indicate that truncation was used.

### HWControlPoint

Host name of the network adapter for the hardware control point. This value will be truncated to seventeen characters. The seventeenth character is a ~ to indicate that truncation was used.

### Status

Indicates if the node is reachable on the network and if the RMC subsystem on the node can communicate with the RMC subsystem on the management server. The valid states are **0** (off), **1** (on) and **127** (unknown). The English representation will be used except when using a delimiter.

### PowerStatus

Indicates the current power status of the node. The valid states are **0** (off), **1** (on), **127** (unknown), and **128** (hardware control not configured). The English representation will be used except when using a delimiter.

### NetworkInterface

Contains the *Name* of the device and the *OpState*.

**Name** The name of the network interface. For example, **eth0** on Linux. Switch Network interfaces are also shown.

### OpState

Represents the current state of the network interface. The valid states are:

- 1** Online
- 2** Offline

## Security

The command requires root access to the cluster management server and a user ID with access to the IBM.NodeHwCtrl resource class in the RMC **ctrmc.acls** ACL file.

This command could require a **systemid** file. For more information, see the **systemid** man page.

## Examples

- The following command returns information in the default format:

```
csmstat
```

```
-----
Hostname           HWControlPoint  Status  PowerStatus  Network-Interfaces
-----
c1sn10.pok.ibm.c~ /dev/tty2      off    off          unknown
c1sn11.pok.ibm.c~ /dev/tty3      off    off          unknown
c1sn12.pok.ibm.c~ /dev/tty4      unknown on         unknown
c1sn13.pok.ibm.c~ /dev/tty4      unknown on         unknown
c1sn14.pok.ibm.c~ /dev/tty4      unknown off        unknown
c1sn15.pok.ibm.c~ /dev/tty4      unknown on         unknown
c1sn16.pok.ibm.c~ /dev/tty4      unknown on         unknown
c1sn17.pok.ibm.c~ /dev/tty4      unknown on         unknown
c1sn18.pok.ibm.c~ /dev/tty4      on     off          en0-Online
```

2. The following command returns information with the specified delimiter:

```
csmstat -d ::
c1sn10.pok.ibm.com::/dev/tty2::0::0::unknown
c1sn11.pok.ibm.com::/dev/tty3::0::0::unknown
c1sn12.pok.ibm.com::/dev/tty4::127::1::unknown
c1sn13.pok.ibm.com::/dev/tty4::127::1::unknown
c1sn14.pok.ibm.com::/dev/tty4::127::0::unknown
c1sn15.pok.ibm.com::/dev/tty4::127::1::unknown
c1sn16.pok.ibm.com::/dev/tty4::127::1::unknown
c1sn17.pok.ibm.com::/dev/tty4::127::1::unknown
c1sn18.pok.ibm.com::/dev/tty4::1::0::en0-1::
```

3. The following command returns information for the specified column headers:

```
csmstat -s Status,Network-Interfaces
```

```
-----
Hostname          Status  Network-Interfaces
-----
c1sn10.pok.ibm.c~ on      en0-Online m10-Offline
c1sn11.pok.ibm.c~ on      sn1-Online sn0-Online at2-Online at1-Online at0-Online
                        en1-Offline en0-Online m10-Offline
c1sn12.pok.ibm.c~ on      en0-Online en1-Offline m10-Offline sn1-Online sn0-Online
c1sn13.pok.ibm.c~ off     unknown
c1sn14.pok.ibm.c~ on      en0-Online en1-Offline at0-Online at1-Online at2-Online
                        at3-Online sn1-Online sn0-Online m10-Offline
c1sn15.pok.ibm.c~ on      en0-Online en1-Offline at0-Online at1-Online at2-Online
                        at3-Online m10-Offline sn1-Online sn0-Online
c1sn16.pok.ibm.c~ unknown unknown
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

**/opt/csm/bin/csmstat**

## See Also

- The **lshwstat** command.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## dcem Command

### Purpose

**dcem** - Opens the DCEM GUI for running a command or script on multiple distributed nodes.

### Synopsis

```
dcem [-h | --help] [-V | --version] [-v | --verbose] [-N | --groups nodegroups]
[-n | --hosts node_list] [command_specification_name]
```

### Description

The **dcem** command opens the Distributed Command Execution Manager (DCEM) GUI for running a command or script on multiple distributed machines on a network simultaneously.

**Note:** This command is not supported for Linux on pSeries clusters. You can specify a collection of individual nodes or create groups of nodes to save and reuse. DCEM provides real-time command status on the individual nodes that you specify, showing them in a waiting, working, successful, or failed state. DCEM helps create, save, and edit command specifications, and it creates a log of all distributed command activity. DCEM provides the capability for saving command specifications as PERL scripts, which can be run simultaneously from the command line. Command specifications are saved under the user's home directory in **\$HOME/dcem/scripts/myscripts**.

The *command\_specification\_name* parameter initializes the input fields with the specified command. This is the name used when the command is saved in the DCEM GUI. The **dsh** command provides the underlying function.

### Parameters

*command\_specification\_name*  
Initializes the input fields with the specified command.

### Options

**-h | --help**  
Writes the command's usage message to standard output.

**-n | --hosts *node\_list***  
Specifies a host name or comma-separated list of host names. Also, overrides the **Host names** field and clears the **Groups of hosts** field.

**-N | --groups *nodegroups***  
Specifies a node group or comma-separated list of node groups. Also, overrides the **Groups of hosts** field and clears the **Host names** field.

**-v | --verbose**  
Runs in debug mode and writes the command's verbose messages to standard output.

**-V | --version**  
Writes version information to standard output.

## Examples

The following examples specify hosts and groups together with the *command\_specification\_name* parameter on the command line. Assume the **myCommand** command was saved with the following host names: **h1**, **h2**, **h3** and groups of hosts: **g1**, **g2**, **g3**.

1. To run DCEM, enter:

```
dcm
```

2. To initialize the input fields with specified command name and groups, enter:

```
dcm -N g4,g5 myCommand
```

This results in the following output in the following GUI fields:

```
Host names:{empty}
Groups of hosts: g4,g5
```

3. To initialize the input fields with specified command name, groups, and hosts, enter:

```
dcm -N g4,g5 --hosts h4 myCommand
```

This results in the following output in the following GUI fields:

```
Host names: h4
Groups of hosts: g4,g5
```

4. To display the DCEM version that is running, enter:

```
dcm -V
```

## Location

`/opt/csm/bin/dcm`

## Files

`$HOME/dcm/scripts/myscripts/script_filename.pl`

Location of the DCEM command specification scripts.

`$HOME/dcm/logs/log_filename`

Location of DCEM log files.

## See Also

- The **dsh** command.
- *IBM CSM for Linux: Administration Guide*.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## definenode Command

### Purpose

**definenode** - Defines the nodes in a CSM cluster.

### Synopsis

**definenode** [-h] [-v | -V] [-m] [-s] [-U] {-f *nodedef\_file*}

**definenode** [-h] [-v | -V] [-m] [--minmanaged] [-s] [-U] {-M *HostnameMapFile*} [-C *ConsoleServerName*] [Attr=value ...]

**definenode** [-h] [-v | -V] [-m] [--minmanaged] [-s] [-U] {[ -n *starting\_node*] [-c *count*] | *node\_list*} [-H *HWControlPoints*] [-C *ConsoleServerName*] [Attr=value ...]

### Description

The **definenode** command runs on the CSM management server to define all the nodes in a CSM cluster. The command creates node definitions in the CSM database. Before running the **definenode** command, you must install CSM on the management server.

This command provides several different options for creating cluster node definitions:

- Create one definition at a time.
- Use the node definition file as input to the command to define multiple nodes.
- Use the **-M** option to automatically define nodes that are listed in a host name mapping file.
- Use the *starting\_node* and *count* values to automatically generate a set of node definitions with consecutive IP addresses.
- Create a node definition file (use the **-s** option and redirect to a file) that can be modified and used as input to the **definenode** command.

#### Defining a Single Node

The information required to define a node can differ depending on variables such as the type of node to be defined, and whether or not hardware control will be supported on the node. The node host name or IP address is always required for all node types. Specific node information is stored in the CSM database and is usually referred to as node attributes. For a complete list of node attributes that can be set using the **definenode** command, see the **nodeattributes** man page.

Default values are set for some attributes if they are not provided at the command line. For example, the default values for the *InstallOSName* and *InstallCSMVersion* attributes are set to the operating system name and CSM version installed on the management server. Also, for nodes with a *PowerMethod* attribute value of *hmc*, the *ConsoleMethod* and *ConsoleServerName* attribute values can also be set automatically. To view the set of values that would be set by the **definenode** command without actually creating the definitions, use the **-s** option.

**Note:** If there are multiple network adapters defined on the management server, then the *ManagementServer* attribute must be set to the management server host name as recognized by the cluster node.

For example, when defining nodes on @server pSeries™ hardware, the following information is usually required:

- The resolvable host name or IP address of the node. This is the nodes' network interface as known by the management server.
- The *PowerMethod* attribute value must be set to **hmc**.
- The *HWControlPoint* and *HWControlNodeId* attribute values.

**Note:** The *HWControlNodeId* value must be unique among all nodes defined in the cluster with the same *HWControlPoint* attribute value.

- The *InstallCSMVersion* and *InstallOSName* attributes default values are the software and version installed on the management server.

For example, the following command defines a node on an HMC-attached pSeries server with host name **clsn01** in a cluster:

```
definenode -n clsn01 -H hmc12 PowerMethod=hmc InstallCSMVersion=1.3.2 \
InstallOSName=Linux
```

For nodes on xSeries servers, the following information is required:

- The resolvable host name or IP address of the node. This is the nodes' network interface as known by the management server.
- The *PowerMethod* attribute must be set to **xseries**.
- The *HWControlPoint* and *HWControlNodeId* attribute values.

**Note:** The *HWControlNodeId* value must be unique among all nodes defined in the cluster with *PowerMethod=xseries*.

- The *ConsoleMethod* attribute must be set to: **mrsv**, **esp**, **computone**, **els**, or **cps**.
- The *ConsoleServerName*, *ConsoleServerNumber*, and *ConsolePortNum* attribute values.
- The *InstallCSMVersion* attribute default value is the software and version installed on the management server.
- The *InstallOSName*, *InstallDistributionName*, *InstallDistributionVersion* and *InstallPkgArchitecture* attributes must be set to appropriate values.

**Note:** The **apc** and **bmc** power methods are also supported for Linux nodes.

To define a node with an RSA (IBM Remote Supervisor Adapter) hardware control point (xSeries), node ID defined as clsn20, and an MRV console server, you would issue a command similar to the following example:

```
definenode -n clsn05 -H rsan3 -C mrsv1::1 PowerMethod=xseries ConsoleMethod=mrsv \
InstallCSMVersion=1.3.2 InstallOSName=Linux HWControlNodeId=clsn20
```

The *HWControlNodeId* attribute is not required. If you do not set the *HWControlNodeId* attribute using the **definenode** command or a **nodedef** file, the node ID will default to the short host name of the node.

### Using a Node Definition File

A node definition file can be used to define multiple nodes. A **nodedef** file consists of stanzas containing node attribute values for each node to be defined; see **/opt/csm/install/nodedef.sample** for an example. This file can be created manually and passed to the **definenode** command using the **-f** option. The **definenode** command then uses the file to create node definitions. When possible, default values are set for attributes that are not specified in the file. The **definenode** command does not accept additional attributes specified on the command line when using the **-f** option.

## definenode

A node definition file can also be created using the **-s** option to redirect the output of the **definenode** command to a file. The file can then be modified as required and passed back to the **definenode** command using the **-f** option.

To define a set of cluster nodes as specified in a node definition file, issue a command similar to the following example:

```
definenode -f /tmp/mynodedefs
```

For more details on the node definition file see the **nodedef** man page.

### Using a Hostname Mapping File

The **-M** option is a convenient way to define large numbers of CSM cluster nodes. The option specifies a file containing a list of host names and associated hardware control information. The **definenode** command uses the host names to create CSM node definitions. Attribute values to use for hardware control, such as *HWControlPoint* and *HWControlNodeId*, are retrieved from the associated hardware control information provided in the file. Any other required attribute values can be specified on the command line.

A host name mapping file can be created by running the **lshwinfo** command, redirecting the output to a file, and editing the file to add the host names for each machine.

For HMC-attached pSeries systems the file could contain information about the LPARs that have been defined. This information could be used to create CSM node definitions by issuing a command similar to the following example:

```
definenode -M /mydir/mylparmap PowerMethod=hmc InstallCSMVersion=1.3.2 \  
InstallOSName=Linux
```

When using xSeries systems the command would look similar to the following example:

```
definenode -M /mydir/mymapfile -C mrv1:0:12 PowerMethod=xseries \  
ConsoleMethod=mrv InstallCSMVersion=1.3.2 InstallOSName=Linux
```

### Defining multiple nodes simultaneously

Multiple nodes can be defined simultaneously, by providing a *starting\_node* and count or by providing one or more node ranges. Using a *starting\_node* and count defines the starting node and increments the node's IP address to determine the value for the next node. IP addresses are incremented up to the count — the total number of nodes. The values for node hardware control and remote console attributes must be specified with the **-H** and **-C** options.

Using a list of node ranges allows greater flexibility in specifying the list of nodes to define. All the nodes in the list of node ranges are defined, and the values for node hardware control, and remote console attributes must be specified with the **-H** and **-C** options. See the **noderange** man page for details on specifying node ranges.

When defining multiple nodes simultaneously, it is assumed that the hardware has been configured in a way that facilitates the use of this method. The mapping of IP address to node hardware and console servers can range in complexity, depending on the type of hardware being used and the cluster environment. Initial planning is required for cluster installation and configuration to be automated as much as possible.

When defining multiple nodes at once, the following is required

- Short host names to be assigned to the nodes are resolvable.



- If using a `starting_node` and `count`, there must be a set of valid consecutive IP addresses that can be assigned to a particular set of nodes.
- The hardware text ID on the ISMP must be set to the short host name of the node. The ISMP (IBM Integrated System Management Processor) device provides monitoring and remote power control for nodes on xSeries servers. ISMPs are also referred to as ASMs.
- A consecutive set of console port numbers must be connected to nodes with consecutive IP addresses. For example, the first port number connects to the first IP address, the second port number connects to the second IP address, and so on.

To define Linux nodes, the `HWControlNodeId` attribute values must be set to the node short host names. The following command would define 30 nodes with consecutive IP addresses as `HWControlNodeId` values `node1` — `node30`:

```
definenode -n node1 -c 30 -s -H rsa1:10,rsa2:10,rsa3:10 -C mrv1::1:20,mrv2::1:20 \
PowerMethod=xseries ConsoleMethod=mrv InstallCSMVersion=1.3.2 \
InstallDistributionName=RedHat \InstallDistributionVersion=8.0
```

Here is the same example using a node range:

```
definenode -n node1+30 -s -H rsa1:10,rsa2:10,rsa3:10 -C mrv1::1:20,mrv2::1:20 \
PowerMethod=xseries ConsoleMethod=mrv InstallCSMVersion=1.3.2 \
InstallDistributionName=RedHat \InstallDistributionVersion=8.0
```

**Note:** The difference between the above examples is that the node range increments by host names, while the `starting_node` and `count` increments by IP address. Depending on the host name resolution, this may result in a different list of nodes being defined.

The nodes are assigned to hardware control points in order up to the total number of nodes for each hardware control point. In the example, there are 10 nodes connected to each hardware control point. The first hardware control point (`rsa1`) is assigned to the first 10 nodes (`node1` - `node10`); the second hardware control point (`rsa2`) is assigned to the next 10 nodes (`node11` - `node20`); the third hardware control point (`rsa3`) is assigned to the last 10 nodes (`node21` - `node30`). The `HWControlNodeId` attributes are set to the node's host names, which should be the same as their ISMP hardware text IDs.

The nodes are assigned to the console servers in order up to the number of ports for each console server. In the example, the first console server (`mrv1`) is assigned to the first 20 nodes (`node1` - `node20`), and the second console server (`mrv2`) is assigned to the next 10 nodes (`node21` - `node30`). The number of console ports must be greater than or equal to the number of nodes being defined. In this case there are only 30 nodes being defined so the last console server is left with some unassigned ports.

In the example, the console port numbers are assigned in consecutive order starting with 1. Port 1 is assigned to the node with the first IP address, port 2 to the next IP address, and so on.

When defining multiple nodes at once, it is important to redirect output to a file and check the file for accuracy before actually creating the node definitions. You can do this using the `-s` option as described earlier.

## definenode

### Parameters

#### *Attr=value*

Specifies a CSM attribute value. For a list of valid attributes to use in the *Attr=value* pairs, see the **nodeattributes** man page. *Attr=value* pairs must be specified last on the command line.

### Options

#### **-C** *ConsoleServerName*

Specifies the list of console server definitions. Multiple console server definitions are separated by commas. The *ConsoleMethod* attribute must be specified when using this option. This information will be used to set the *ConsoleServerName*, *ConsoleServerNumber*, and *ConsolePortNum* attributes of the node definition. The entries must have the following format:

```
-C console_server_name[:console_server_num[:starting_port_num \
[:num_console_ports]]][, ...]
```

where:

#### *console\_server\_name*

Specifies the host name or IP address of the console server.

#### *console\_server\_num*

Specifies the console server number. The default is 1. This attribute applies to the **esp** console method only.

**Note:** CSM supports the ESP console server on Red Hat 7.2, 7.3, and 8.0 management servers only.

#### *starting\_portnum*

Specifies the starting console port number. The default is 0 for the **esp** method and 1 for other methods. If some ports have already been assigned, then a starting number greater than the default must be specified. For the **esp** method this value is a hexadecimal (0-f); for other methods the value is a decimal.

#### *num\_console\_ports*

Specifies the total number of ports on the console server. Each node is assigned a specific port number and console server. The console servers are used in the order they are specified. All the port numbers are assigned for the first console server, and then ports are assigned from the next console server, and so on. If one of the ports is already in use then the **definenode** command will not complete successfully and no nodes will be defined.

**Note:** The **-C** flag cannot be used when specifying the **csp** *PowerMethod*.

#### **-c** *count*

Specifies how many nodes to define. The IP addresses are determined by incrementing the starting node IP address by one. To use this option nodes must be defined with consecutive IP addresses and entered into the nameserver consecutively. This option is not valid when supplying a node range.

#### **-f** *nodedef\_file*

Specifies the node definition file name. A sample node definition file is

supplied in `/opt/csm/install/nodedef.sample`. See “Using a Node Definition File” on page 47 for more information. This option cannot be used with any other option.

**-h** Writes the command’s usage statement to standard output.

**-H *HWControlPoints***

Specifies the list of hardware control points. Multiple hardware control point definitions are separated by commas. The *PowerMethod* attribute must be specified to set the *HWControlPoint* and *HWControlNodeId* attributes of the node definition. Each definition must be specified using the following format:

```
-H HWCtrlPt[:num_nodes][, ... ]
```

where:

*HWCtrlPt* Specifies a hardware control point by host name or IP address.

*num\_nodes* Specifies the total number of nodes controlled by this hardware control point. The default is 1.

**-m** Modifies existing node definitions or creates new node definitions. If you run **definenode** on a node that has already been defined, using the **-m** option will change those definitions. This is helpful if you define your nodes in a node definition file, run **definenode -f *nodedef\_file***, and then want to change the definitions for several nodes. Instead of running **chnode**, you can make the changes in the *nodedef\_file* file and run **definenode -mf *nodedef\_file***. This updates the CSM database to contain the changes made in the node definition file. The **-m** option can be run with any other options. If the **-m** option is specified but there are no existing node definitions, then **definenode** will create the node definitions specified.

**--minmanaged**

Sets the *Mode* attribute to **MinManaged** for nodes that should have no CSM or RSCT code installed. Use this option to define multiple Minimally Managed nodes in the CSM database prior to installing CSM.

**-M *HostnameMapFile***

Specifies the name of a file that contains a mapping of network IP addresses to hardware control information. This file is initially created by redirecting the output of the **lshwinfo** command. The file then must be edited to assign host names to the individual machine descriptions. To use this option the *PowerMethod* attribute of the node definition must be specified. This option cannot be used with the **-f**, **-n**, **-c**, or **-H** options. If the *PowerMethod* is **hmc** or **csp** then the hardware control and console attributes are set automatically. For other power methods the hardware control information is also set automatically, but the console information must be provided.

**-n *starting\_node* | *node\_list***

When *starting\_node* is supplied, this option specifies the IP address or host name for the first node to be added. A list of nodes is generated by incrementing IP addresses up to the value specified by the **-c *count*** value.

When *node\_list* is supplied, this option specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes.

For information about specifying node ranges, see the **noderange** man page.

## definenode

- s** Sends the list of node definitions that the **definenode** command would create to standard output. This output can be directed to a file to use as a node definition file.
- U** Runs the **updatenode** command to install the nodes once they are defined.
- v | -V** Writes the command's verbose messages to standard output.

## Exit Status

- 0** The command has run successfully.
- 1** An error has occurred.

## Security

The command requires root access to the cluster management server.

## Examples

- 1. To define a machine as a cluster node using the default attribute values (in this case the LPAR name is not the same as the node short host name), enter:  

```
definenode -n node12 -H hmc01 HWControlNodeId=1par12 PowerMethod=hmc
```
- 2. To define a node on xSeries hardware (in this case the text ID is not the same as the node short host name), enter:  

```
definenode -n node12 -H xseries01 HWControlNodeId=1par12 PowerMethod=xseries \  
InstallOSName=Linux InstallDistributionName=RedHat \  
InstallDistributionVersion=8.0 InstallPkgArchitecture=i386
```
- 3. To create node definitions for each node described in the node definition file **/tmp/mynodedefs**, enter:  

```
definenode -f /tmp/mynodedefs
```
- 4. To create cluster node definitions for all nodes listed in the host name mapping file **/u/username/nodemap1** using the default attribute settings, enter:  

```
definenode -M /u/username/nodemap1 PowerMethod=xseries
```
- 5. To create a node definition file by redirecting the output of the **definenode** command to a **nodedefs** file, enter:  

```
definenode -s -M mymapfile PowerMethod=xseries InstallCSMVersion=1.3.2 \  
InstallOSName=Linux InstallDistributionVersion=5.2.0 > nodedefs
```

## Implementation Specifics

This command is part of the Cluster Systems Management (CSM) software.

## Location

**/opt/csm/bin/definenode** Location of the **definenode** command.

## Files

**/opt/csm/install/nodedef.sample**  
Location of the sample node definition file.

## See Also

- The **chnode**, **lshwinfo**, **lsnode**, **nodedef**, **rmnode**, and **updatenode** command man pages.
- The **nodeattributes** man page.
- The *IBM CSM for Linux: Planning and Installation Guide*.
- The *IBM CSM for Linux: Hardware Control Guide*.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## dsh Command

### Purpose

**dsh** - Concurrently issues remote shell commands to multiple hosts and formats results.

### Synopsis

**dsh -q**

```
dsh [-a] [-c] [-i ] [-h] [-m] [-q] [-v] [-s] [-z] [-S csh | ksh] [-L] [-I login_name]
[-n node_list | -] [-w [node_list] | - ] [-N nodegroups] [-o "remote_shell_options"]
[-r remote_shell_path] [-f fanout_value] [-t timeout] [-R | --reports directory] [--log
pathname/filename] [-W | --noFileWriting] [command]
```

### Description

The **dsh** command invokes commands on a set of nodes concurrently. It issues a remote shell command concurrently for each node that is specified and returns the output from all the nodes, formatted so that command results from all nodes can be managed. **/usr/bin/rsh** is the model for syntax and security.

The set of nodes to which commands are sent can be determined in two ways:

The preferred method is the **node list**. The node list is obtained from the first instance of one of the following:

1. A list of host names is specified on the command line when the **-n** option is used. A list of node groups is specified on the command line when the **-N** option is used.
2. The contents of a file named by the **DSH\_LIST** environment variable. The node list file format is one host name per line. Blank lines and comment lines beginning with **#** are ignored.

Another method is called the **working collective**. The working collective is obtained from the first instance of one of the following:

1. A list of host names is specified on the command line when the **-w** option is used. A list of node groups is specified on the command line when the **-N** option is used.
2. The contents of a file named by the **WCOLL** environment variable. The working collective file format is one host name per line. Blank lines and comment lines beginning with **#** are ignored.

An error occurs if neither a node list nor a working collective exists when this method is used, and no commands are issued.

If nodes are specified in more than one way, only the highest priority specification is used, as follows:

1. **-n** option
2. **-w** option
3. **DSH\_LIST**
4. **WCOLL**

If the *command* parameter is not specified, **dsh** reads lines from the command line or standard input and issues each input as a command on each host in the node list or working collective. The commands use the syntax of the remote shell command.

To exit the **dsh** command line mode, enter **exit** on the command line or press the **Enter** key at the **dsh** prompt.

When commands are resolved on the remote node, the path used is determined by the **DSH\_PATH** environment variable specified by the user. If **DSH\_PATH** is not set, the path used is the remote shell default path. (For example, to set **DSH\_PATH** to the path set on the source node, use **DSH\_PATH=\$PATH**).

**dsh** runs commands directly on the local host rather than using the remote shell unless the **-l** option is used.

The maximum number of concurrent remote shell commands can be specified with the fanout **-f** option or using the **DSH\_FANOUT** environment variable. If desired, sequential invocation can be obtained by specifying a fanout value of 1. The fanout is kept at the fanout number that is specified. When one command is completed on a node, another command is started. If fanout is not specified by the **DSH\_FANOUT** environment variable or by the **-f** option, then a default fanout of 64 is used. Each remote shell command that **dsh** runs requires a reserved TCP/IP port, and only 512 such ports are available per node.

If the streaming mode is specified by the **-s** option instead of the fanout mode, then output is returned from each node as the command is completed on that node, instead of waiting for the command to be completed on all nodes before the results are returned. This can improve performance but causes the output to be unsorted.

Exit values for the remote shell commands are displayed in messages from the **dsh** command if the exit values are nonzero. A nonzero return code from a remote shell indicates that the remote shell has encountered a problem. This is unrelated to the exit code of the remotely issued command. If a remote shell encounters a problem, that node is removed from the current node list. Use the **-z** option to obtain the return code from the last command issued on the remote node. Note that OpenSSH behaves differently; it returns the exit status of the remote command as its exit status. This affects the behavior of **dsh** and may require the use of the **-c** option.

The **dsh** exit value is 0 if no errors occurred in the **dsh** command and all remote shell commands finished with exit codes of 0. If internal errors occur or the remote shell commands do not complete successfully, the **dsh** exit value is greater than 0. The exit value is increased by 1 for each successive instance of an unsuccessful run.

No particular error recovery for command errors on remote hosts is provided. The application or user can examine the command results in the standard error and standard output of the **dsh** command and take appropriate action.

The **dsh** command waits until results are in for each command for all hosts and displays those results before reading more input commands. This is true only if the **-s** option is not specified on the **dsh** command line.

The **dsh** command does not work with interactive commands, including those read from standard input.

## dsh

The **dsh** command output consists of the output (standard error and standard output) of the remotely issued commands. The **dsh** standard output is the standard output of the remote shell command. The **dsh** standard error is the standard error of the remote shell command. Each line is prefixed with the host name of the node which produced the output. The host name is followed by ":" and a line of the command output.

For example, a command was issued to a node list of host1, host2, and host3. When the command was issued on each of the hosts, the following lines were written by the remote commands:

```
For host1 stdout:
h1out1
h1out2

For host2 stdout:
h2out1
h2out2

For host3 stdout:
h3out1

For host3 stderr:
h3err1
h3err2

dsh stdout will be
host1: h1out1
host1: h1out2
host2: h2out1
host2: h2out2
host3: h3out1

dsh stderr will be
host3: h3err1
host3: h3err2
```

A filter to display identical outputs grouped by node is provided separately. See the **dshbak** command.

If a node is detected as being down (for example, a remote shell command issues a non-zero return code), subsequent commands are not sent to this node on this invocation of the **dsh** command unless the **-c** option is specified.

An exclamation point (!) at the beginning of a command line causes the command to be passed directly to the local host in the current environment. The command is not sent to the node list.

Signal 2 (INT), Signal 3 (QUIT), and Signal 15 (TERM) are propagated to the remote commands.

Signal 19 (CONT), Signal 17 (STOP), and Signal 18 (TSTP) are defaulted. This means that the **dsh** command responds normally to these signals, but the signals do not have an effect on the remotely running commands. Other signals are caught by **dsh** and have their default effects on the **dsh** command. In the case of these other signals, all current child processes, and by means of propagation their remotely running commands, are terminated (SIGTERM).

**Note:** The **DSH\_REMOTE\_CMD** environment variable can be used to specify a remote shell other than the default **/usr/bin/rsh**. For example, a secure



remote command that conforms to the IETF (Internet Engineering Task Force) secure remote command protocol. Be aware, however of the following limitations:

1. The **dsh** itself has no security configuration or obligations. All security issues are related to the remote environment enabled by the user and the security configuration level that the user has implemented. For example, if the remote shell requires public keys, it is the responsibility of the user to implement this.
2. Use the fully qualified host name when you define a node for the remote shell. If the remote shell requires a list of nodes in its configuration, then the nodes must be defined by their fully qualified host names. This allows the **dsh** command to recognize the node. You can also use an alias to define a node. Aliases are permitted provided the fully qualified host name is also provided.

You can specify the remote shell you wish to use in the following ways. The order shown here is the order of precedence.

- **-r** option
- **DSH\_REMOTE\_CMD** environment variable
- CSM *RemoteShell* attribute (see the **csconfig** command).

If none of the above are specified, **dsh** defaults to **/usr/bin/rsh**.

Note that **dsh** defaults to ksh syntax unless you use the **-S** option.

## Parameters

*command* Specifies a command to invoke on the node list. It is passed to the remote shell. This command is specified by using the remote shell command syntax.

## Options

- a** Adds all nodes defined to CSM to the node list. Note that you cannot use the **-a** option with the **-n** or **-w** option.
- c** Specifies that commands that were unsuccessful should continue to be sent to the remote nodes.
- f** *fanout\_value* Specifies a fanout value. The default value is 64. It indicates the maximum number of concurrent remote shell commands to issue. Sequential runs can be specified by indicating a fanout value of 1. The fanout value is taken from the **DSH\_FANOUT** environment variable if the **-f** option is not specified.
- h** Writes the command's usage statement to standard output.
- i** Informs the user that a node is not responding and prompts the user to decide whether the node should be included in the node list.
- l** *login\_name* Specifies a remote user name under which to invoke the commands. If **-l** is not used, the remote user name is the same as the local user name. Use this option as you would with the remote shell command.
- log** *pathname/filename* Sets the log file. Log entries are added for each instance of the **dsh** command. This option overrides the **DSH\_LOG** environment variable.

## dsh

- L** Specifies to not export locale environment.
- m** Prints the results of monitoring for each node in the form of the starting and completion messages for each node.
- n *node\_list* | -**

Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

If **-** is specified, you enter standard input mode. **dsh** prompts for host names with **dsh host>".** Enter the host names one line at a time. When you are finished, press **<Ctrl+d>** to exit standard input mode and return to the **dsh** prompt. If **-n -** is used, commands cannot be read from standard input when redirected from a file.

**Note:** Duplicate host names are included only once in the node list.
- N *nodegroups***

Resolves one or more CSM node groups, separated by commas, and adds the nodes to the node list or working collective.
- o "*remote\_shell\_options*"**

Forwards options for the remote shell. The information within the quotation marks is forwarded and included in the remote shell.
- q** Displays the current environment variable settings. For example, the list of nodes in the current node list or working collective file and the value of the **DSH\_FANOUT** environment variable are displayed.
- Note:** This option must be specified by itself on the **dsh** command line. It cannot be used in conjunction with any other **dsh** option or with the *command* parameter.
- r *remote\_shell\_path***

Provides the full path of the remote shell that is used to access the remote systems. The default remote shell is **rsh**.
- R | --reports *directory***

Sets the reports directory for reporting. All output from each host is saved to individual output and error files and a report is generated. A unique subdirectory is created to hold the output files and the report for each instance of the command. The subdirectory name is created using the default **unspecified** or the string provided using the **--reportName** option. An integer extension is added to the name to create a unique directory name. The option overrides the **DSH\_REPORTS\_DIRECTORY** environment variable.
- s** Specifies output in streaming mode. The output is unsorted, but performance is likely to improve, and memory utilization is reduced.
- S *cs* | *ks***

Specifies the shell syntax to be used on the remote system.
- t *timeout***

Specifies the time, in seconds, to wait for response from the remote shell. Overrides the value of the **DSH\_TIMEOUT** environment variable.
- v** Verifies a node before adding it to the node list. If a node is not responding, it is not included in the node list. If the **/opt/csm/bin/lsnode** command is installed, then it can be used to check the **ping** status of the node. If the

**Isnode** command is not installed or if the status returned is not zero, then the command **/bin/ping** can be used to check the node. The **/bin/ping** command takes a few seconds to check the node that is not responding, rather than the minute typically taken for the remote shell command to time out.

**-w node\_list | -**

Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. Host names must be specified as positional arguments or preceded by the **-w** flag. For information about specifying node ranges, see the **noderange** man page.

If **-** is specified, you enter standard input mode. **dsh** prompts for host names with **dsh host>".** Enter the host names one line at a time. When you are finished, press **<Ctrl+d>** to exit standard input mode and return to the **dsh** prompt. If **-w -** is used, commands cannot be read from standard input when redirected from a file.

**Note:** Duplicate host names are included only once in the working collective.

**-W | --noFileWriting**

Disables output file writing for an instance of the command when used with the **--reports** option or when the **DSH\_REPORTS\_DIRECTORY** is set. Reporting remains enabled. This option is required when using reporting with the **dshbak** command.

**-z**

Prints the return code of the last command that was run remotely. The return code is appended at the end of the output for each node.

## Environment Variables

### **DSH\_FANOUT**

Sets the maximum number of concurrent remote shell commands. This can also be set by the **-f** option.

### **DSH\_LIST**

Specifies a file that contains definitions of the set of nodes that comprise the node list.

### **DSH\_LOG**

Enables logging when set to the absolute path and file name of the log file. This variable is overridden by the **--log** and **--noLogging** options.

### **DSH\_PATH**

Sets the path that is used on the remote nodes. If **DSH\_PATH** is not set, the default path for the remote shell is used. For example, **DSH\_PATH=\$PATH** sets the path on the remote node to the same path that is used on the source node.

### **DSH\_REMOTE\_CMD**

Specifies the path of the remote shell executable to use instead of the default.

### **DSH\_REMOTE\_OPTS**

Includes the options specified in the remote command when the command is forwarded to the remote nodes.

## dsh

|  
|  
|  
|

### DSH\_REPORTS\_DIRECTORY

Enables reporting and output file writing when set to the reports directory path. This variable is overridden by the **--reports** and the **--noReporting** options.

### DSH\_TIMEOUT

Specifies the time, in seconds, to wait for response from the remote shell. This can also be set with the **-t** option.

### WCOLL

Specifies a file that contains definitions of the set of nodes that comprise the working collective.

## Security

The command requires root access to the cluster management server. Additional security considerations are the same as those for the remote shell command.

## Examples

1. To run the **ps** command on each host listed in the **dshhosts** file, enter:  

```
DSH_LIST=./dshhosts dsh ps
```
2. To list the current node list file as specified by the **DSH\_LIST** environment variable, enter:  

```
dsh -q
```
3. To set the node list to three nodes and start reading commands from standard input, enter:  

```
dsh -n otherhost1,otherhost2,otherhost3
```
4. To set the current node list to three nodes and issue a command on those nodes while formatting the output, enter:  

```
dsh -n host1,host2,host3 -a cat /etc/passwd | dshbak
```
5. To append the file **remotefile** on the node named **otherhost** to the file named **otherremotefile**, which is located on **otherhost**, enter:  

```
dsh -n otherhost cat remotefile '>>' otherremotefile
```
6. To run a file of commands sequentially on all the members of the current working collective and save the results in a file, including the collective and the working collective for each command, enter:  

```
dsh -if 1 <commands_file >results 2>&1
```
7. To run the **ps** command on the node list and filter results locally, enter:  

```
dsh ps -ef | grep root
```
8. To run the **ps** command and filter results on the node list hosts (this can improve performance considerably), enter:  

```
dsh 'ps -ef | grep root'
```

or

```
dsh ps -ef "|" grep root
```
9. To **cat** a file from **host1** to the local system, stripping off the preceding host name to preserve the file, enter:

```
dsh -n host1 cat /etc/passwd | cut -d: -f2- | cut -c2- >myetcpasswd
```

10. To run the **needs\_auth\_program** with the **-D** option specified on the remote shell on all of the nodes in the cluster, enter:

```
dsh -a -o "-D" /usr/bin/needs_auth_program
```

11. To enter a list of host names in standard input mode by specifying **-n-** and then request the date from the specified nodes, enter:

```
dsh -n -
```

When you complete the list of host names, press **<Ctrl+d>** to return to the **dsh** prompt. At the **dsh** prompt, specify:

```
date
```

Output will be similar to the following:

```
# dsh -n -
host1
host2
host3
dsh> date
host1: Fri Mar 23 08:46:59 EST 2001
host2: Fri Mar 23 08:46:59 EST 2001
host3: Fri Mar 23 08:46:59 EST 2001
dsh> exit
#
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

### **/opt/csm/bin/dshbak**

Location of the command that is supplied as the back-end formatting filter.

### **node list file**

File that contains host names, one per line, that defines a set of nodes which comprise the node list. This file is specified by the **DSH\_LIST** environment variable.

### **working collective file**

File that contains host names, one per line, that defines a working collective. This file is specified by the **WCOLL** environment variable.

## Location

**/opt/csm/bin/dsh**

## See Also

The **dshbak** and **rsh** commands.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## dshbak Command

### Purpose

**dshbak** - Presents formatted output from the **dsh** command.

### Synopsis

**dshbak** [-c]

### Description

The **dshbak** command formats output from the **dsh** command. The **dshbak** command takes lines in the following format:

```
host_name: line of output from remote command
```

The **dshbak** command formats the lines as follows and writes them to standard output. Assume that the output from `host_name3` and `host_name4` is identical, and the **-c** option was specified:

```
HOSTS -----
host_name1
-----
.
.
lines from dsh with host_names stripped off
.
.
HOSTS -----
host_name2
-----
.
.
lines from dsh with host_names stripped off
.
.
HOSTS -----
host_name3          host_name4
-----
.
.
lines from dsh with host_names stripped off
.
.
```

When output is displayed from more than one node in collapsed form, the host names are displayed alphabetically. When output is not collapsed, output is displayed sorted alphabetically by host name. The **dshbak** command writes "." for each 1000 lines of output filtered.

### Parameters

None.

### Options

**-c** Collapses identical output from more than one node so that it is displayed only once.

### Security

The command requires root access to the cluster management server.

## Examples

1. To display the results of a command issued on several nodes, in the format used in the Description section above, enter:

```
dsh -n node1,node2,node3 cat /etc/passwd | dshbak
```

2. To display the results of a command issued on several nodes with identical output displayed only once, enter:

```
dsh -w host1,host2,host3 pwd | dshbak -c
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

**/opt/csm/bin/dshbak**

When the **dshbak** filter is used and standard error messages are generated, all error messages on standard error appear before all standard output messages. This is true with and without the **-c** option.

## See Also

The **dsh** command.

## Implementation Specifics

IBM CSM Development - cluster@us.ibm.com

## getadapters Command

### Purpose

**getadapters** - Collects MAC addresses for network adapters installed on a node.

### Synopsis

```
getadapters [-h] [-w] [-o filename] [-m mac_method] [-t type [-D [-A] -s auto |
10 | 100 | 1000 -d auto | half | full [-S server_ip] [-G gateway_ip]]] [-m
mac_method] {[-f filename | -a | [-n node_list] [-N nodegroups] } [Attr=value ...]
```

### Description

The **getadapters** command collects the MAC address of the first Ethernet adapter installed in a node. The command uses the **dsh** MAC method to get a MAC address. If the node is not in **Managed** or **MinManaged Mode**, or if **dsh** is not reachable, then it uses the **pxeboot** method for xSeries, IntelliStation, and eServer 325, the **hwstat** method for BladeCenter, and the **hmc** method for pSeries. The **getadapters** command powers down the target nodes to obtain this information when using the **pxeboot** method, so when the command completes the nodes must be rebooted.

If the **-f** option is provided, the specified file name is used as an input file for the **getadapters** command. The **getadapters** command will read the adapter information from the file rather than powering off the nodes and acquiring the information. The input format of the file is as follows:

```
Name::Adapter Type::MAC :: :: ::
```

**Note:** The command is currently limited to running on a maximum of eight nodes at a time.

For nodes on HMC-attached pSeries servers, SP Nodes, and p660 servers, the list of adapters returned can be restricted to a particular type by using the **-t** option. The **-t** option accepts the **ent** (Ethernet), **tok** (Token Ring), **fddi** (FDDI), or **sni** (IBM eServer pSeries High Performance Switch) adapter attributes to specify the adapter type.

For nodes on HMC-attached pSeries servers, adapter discovery can be requested using the **-D** option. For discovery, the **getadapters** command will perform a ping test from the node to a destination address to verify the network path. The adapter type is required for discovery, so the **-t** option must be provided. By default, only the first adapter of the specified type that successfully pings the destination address will be returned. The **-A** option can be used with discovery to request that all adapters of the specified type that successfully ping the server be returned. If the adapter type is Ethernet, the network speed and duplex must also be provided. The **-s** option is used to provide the adapter speed, which can be **auto**, **10**, **100**, or **1000**. The **-d** option is used to provide the adapter duplex, which can be **auto**, **half**, or **full**. If the adapter type is Token Ring or FDDI, the network speed and duplex are not used.

An output file can be specified with the **-o** option. If an output file is specified, adapter information that would by default be written to the screen is written to the specified file instead. Any informational or error messages are still written to the screen. The output file can be edited and used as an input file with the **-f** option for another run of the command.



Certain data returned from the command may be written to the CSM database. The **-w** option writes the MAC address to the CSM database *InstallAdapterMacaddr* attribute for the specified node. If a network gateway is specified with the **-G** option or from an input file, this value is written to the CSM database *InstallAdapterGateway* attribute for the specified node. If more than one adapter is returned for a node, only information for the first adapter returned is written to the CSM database.

**Notes:**

1. Only one of the **-a**, **-f**, **-n**, or **-N** options can be specified.
2. To automatically obtain MAC addresses from x445 servers using **getadapters** or **csmssetupks**, the management server must be running the 2.4.9-e.24smp or 2.4.9-e.24summit kernel.

## Parameters

*Attr=value ...*

Specifies one or more *Attr=value* pairs separated by spaces. *Attr=value* pairs must be specified last on the command line. Valid attributes are:

*Gateway*

Specifies a global default gateway to use in the **/etc/dhcpd.conf** file for installation. If there are additional gateways (routers) on the management server for different subnets to which individual nodes are connected, the individual gateways are determined automatically by CSM.

*Nameservers*

Specifies the nameservers to use in the **dhcpd.conf** file and in the Kickstart configuration file.

## Options

**-a** Processes all nodes in the CSM database. You cannot use the **-a** option with the **-n** option.

**-d auto | half | full**

Specifies Ethernet network duplex. This option is not used for Token Ring or FDDI adapters. This option is not valid for Linux; using it will cause an error message and the command will not complete successfully.

**-D** Discovers the first adapter that can reach the server.

**-f** Specifies to read adapter information from the specified file, instead of acquiring the information directly from the nodes.

**-h** Displays usage information.

**-m mac\_method**

Specifies the type of MAC method to use to collect a MAC address. The supported methods are **dsh**, **pxeboot**, and **hwstat**. You can also specify your own MAC method. User scripts should return data with values space-delimited, in the following format (one line for each node).

```
MAC_DATA:node_name adapter_type physical_location \  
MAC_address adapter_speed adapter_duplex
```

For example:

```
MAC_DATA:blade2.clusters.com eth0 NA 00:09:6B:00:20:51 NA NA
```

## getadapters

Only the *node\_name*, *adapter\_type*, and *MAC\_address* are required. All others can have a value of **NA** if they do not apply to the MAC method.

**-n** *node\_list*

Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

Note that you cannot use the **-n** option with the **-a** option.

**-N** *nodegroups*

Specifies the node groups to process.

**-o** *filename*

Writes adapter information to the specified file instead of to the screen.

**-s**

Specifies the Ethernet network speed. This option is not used for Token Ring or FDDI adapters. This option is not valid for Linux; using it will cause an error message and the command will not complete successfully.

**-S** *server\_ip*

Specifies the IP address or host name of the server to discover. The default is the management server. This option is not valid for Linux; using it will cause an error message and the command will not complete successfully.

**-t** *type*

Returns only adapters of the given type. This option is not valid for Linux; using it will cause an error message and the command will not complete successfully.

**-w**

Writes adapter attributes to the CSM database.

## Exit Status

- 0** The command completed successfully.
- 1** The command did not complete successfully.

## Security

The command requires root access to the cluster management server.

This command may require a **systemid** file. For more information, see the **systemid** command.

## Examples

1. To list all adapters in all nodes and write the information to the CSM database (only information for the first adapter found in each node will be written to the database), enter:  

```
getadapters -a -w
```
2. To list all adapters in select nodes, enter:  

```
getadapters -n node1,node2,node4
```
3. To read the adapter information from the **adapters\_file** file, in the current directory, and write that information to the CSM database, enter:  

```
getadapters -f ./adapters_file -w
```

## **Implementation Specifics**

This command is part of IBM Cluster Systems Management (CSM) software.

## **Files**

**`/var/log/csm/getadapters/*`**

Location of the log output from the **getadapters** command.

## **See Also**

- The **csmsetupks** and **csmsetupyast** commands.
- The *IBM CSM for Linux: Planning and Installation Guide*.

## **Location**

**`/opt/csm/bin/getadapters`**

## **Author**

IBM CSM Development - cluster@us.ibm.com

---

## installms Command

### Purpose

**installms** - Installs the Cluster Systems Management (CSM) management server software.

### Synopsis

**installms** [-h]

**installms** [-f] [-v | -V] [-c] {-x | -p *pkg\_path:[pkg\_path...]*} [Attr=value ...]

### Description

The **installms** command installs the CSM management server software. It automatically installs both CSM and its prerequisite software. Before running the **installms** command, a supported version of the operating system must be installed.

The command performs the following tasks:

- Creates the **/csminstall** directory structure that is required for CSM files.

**Note:** If the **/csminstall** directory does not exist, it will be created by the **installms** command. However, it is recommended that you create the **/csminstall** directory in a separate partition before running the **installms** command. See the *CSM for Linux: Planning and Installation Guide* for more information on requirements for **/csminstall**.

- Optionally copies the required CSM, open source and distribution RPM packages to the correct **/csminstall** subdirectories. Note that if the **installms** command cannot find the required RPM packages, it will prompt you to insert the Linux distribution CD-ROMs.
- Copies the CSM scripts and files that are required to install CSM on the nodes of the cluster to the correct **/csminstall** subdirectories.
- Sets global cluster system attribute values.

If the **-p** option is used, the **installms** command copies the required packages from the specified location into the correct subdirectories of **/csminstall**. If the **-x** option is used, the **installms** command assumes that the files have already been copied. When the **installms** command installs the management server software, it uses the packages that have been copied to the **/csminstall** subdirectories. For a list of required packages, see the *IBM CSM for Linux: Planning and Installation Guide*.

### Parameters

*Attr=value ...*

Specifies one or more *Attr=value* pairs separated by spaces. *Attr=value* pairs must be specified last on the command line. Valid attributes are:

*RemoteShell=full\_path\_name\_of\_shell*

The *RemoteShell* attribute value determines the default remote shell for the **dsb** command if the **DSH\_REMOTE\_CMD** variable is not set. It also determines which shell to set up on the nodes during installation. The default attribute value for the management server is **/usr/bin/ssh**.

*SetupRemoteShell={yes | no}*

The *SetupRemoteShell* attribute value controls whether the remote

shell is set up on the management server and the nodes. If it is set to **yes** or **1**, then the remote shell is set up. The remote shell is not set up if the attribute is set to **no** or **0**. The default value for this attribute is **1**.

## Options

- c** Forces the command to copy all required files from the *pkg\_path*. Normally, **installms** only copies files that are newer or the same as those in **/csminstall**, or files that do not exist in **/csminstall**. The **-c** option forces **installms** to copy all files regardless of their version.
- f** Forces installation regardless of the status of the existing installation.
- h** Writes the command's usage statement to standard output.
- p** *pkg\_path*  
Specifies one or more colon-separated directories where installation packages are located. This option cannot be used with **-x**.
- v | -V**  
Writes the command's verbose messages to standard output.
- x** Specifies to not copy software packages and to use packages in the **/csminstall** subdirectory. This option cannot be used with **-p**.

## Exit Status

- 0** The command has run successfully.
- 1** An error has occurred.

## Security

The command requires root access to the cluster management server.

## Examples

1. To install CSM and the packages contained in the current directory on the management server using the force option, enter:  
`installms -f -p .`

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

### **/var/log/csm/installms.log**

Location of the log file on the management server for the **installms** command. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4.

## Location

**/opt/csm/bin/installms**

## See Also

- The **addlcsext**, **csmconfig**, **definnode**, **uninstallms**, and **updatenode** commands.

## **installms**

- The *IBM CSM for Linux: Planning and Installation Guide*.

## **Author**

IBM CSM Development - cluster@us.ibm.com

## installnode Command

### Purpose

**installnode** - Installs the Linux operating system and CSM on nodes in a Linux cluster.

### Synopsis

```
installnode [-h] [-v | -V] {-P | -a | [-N nodegroups] [[-n] node_list] [--file filename]}
```

### Description

The **installnode** command runs on the management server to perform a full installation on the specified nodes. The **installnode** command installs the Linux operating system and CSM.

If given no arguments, the **installnode** command displays its usage statement. This prevents an unintended installation of all nodes. To install all nodes whose *Mode* attribute is **PreManaged**, use the **-P** option. To install all nodes, use the **-a** option. To install a specific set of nodes or node groups, specify the **-N** option or a list of nodes. Both a list of nodes and a list of node groups may be provided in the same invocation of the **installnode** command.

The **installnode** command installs only Linux nodes whose *InstallMethod* attribute is set to **kickstart** or **autoyast**. All other nodes are skipped and a warning is displayed. The *InstallMethod* attribute is set using the **csmsetupks** or **csmsetupyast** commands, or can be changed using the **chnode** command.

CSM is not installed on Minimally Managed nodes (*Mode=MinManaged*). Only the operating system will be installed on these nodes. In addition, user-provided customization scripts will be run on Minimally Managed nodes, and the remote shell will be set up. Run the **monitorinstall** command on the management server to display installation status of Minimally Managed nodes

The following log files provide a history of the installation process. The **installnode.log** is located on the management server in **/var/log/csm**. This log records the verbose output of the **installnode** command. A log is also created on each node in **/var/log/csm/install.log**. This log records in detail what happened to the node as it was installed. For Linux nodes on pSeries servers, the **/var/log/csm/installnode..log.\*** script on the management server will recode **hmc\_nodecond** output for the **installnode** command. The **monitorinstall** command also displays installation status.

The **installnode** command also runs user-provided customization scripts on the nodes that are being installed. Any scripts in **/csminstall/csm/scripts/installpreboot** are automatically run on the nodes after the operating system has been installed, but prior to the first reboot of the node. Any scripts in **/csminstall/csm/scripts/installpostreboot** are automatically run on the nodes after the first reboot of the node and after CSM has been installed. For more information on CSM support for user-provided customization scripts see the *IBM CSM for Linux: Administration Guide*.

As each node is installed, it is added to the cluster and becomes a Managed node.

## installnode

Before the **installnode** command can be run, the following prerequisites are required:

1. NFS must be available on the management server for mounting **/csminstall** and **/tftpboot** to the nodes.
2. The **installms**, **definnode**, and **csmsetupks** or **csmsetupyast** commands must have already been run.
3. Before running a full installation, set each node's boot order to the following:
  - Floppy
  - CD-ROM
  - Network
  - Hard disk

During a Kickstart installation, the **installnode** command completes the following steps:

1. Sets up the **/tftpboot/pxelinux.cfg/node-ip-addr-in-hex** file for each node, so that when a node reboots it does a Kickstart installation.
2. Reboots the node.
3. As the node reboots, broadcasts its MAC address.
4. The **dhcp** server on the management server accepts the node's **dhcp** request, and **pxelinux** initiates a Kickstart installation.
5. Kickstart installs the operating system on each node.
6. After the Kickstart installation of the operating system is complete, the Kickstart post-installation script, **/csminstall/csm/InstallCSMVersion/kickstart.InstallDistributionNameInstallDistributionVersion/node-ipaddr-kickstart**, is run on the node. After some additional configuration, any user scripts in **/csminstall/csm/scripts/installprereboot** are run. Then the Kickstart post-installation script adds the **csmfirstboot** script **/csminstall/csm/InstallCSMVersion/csmfirstboot** to **/etc/inittab**. It then modifies the **pxelinux** configuration file **/csminstall/pxelinux.cfg/node-ip-addr-in-hex** on the management server so that the next reboot of the node reboots from the local hard drive.
7. The node reboots to the local hard drive.
8. As the reboot completes, the **csmfirstboot** script, which was listed in **/etc/inittab**, is run on the node.
9. The **csmfirstboot** script runs **makenode** and any user scripts in **/csminstall/csm/scripts/installpostreboot**. The **makenode** command installs CSM and its dependencies and sets its management server. Then **csmfirstboot** removes itself from **/etc/inittab**.

**Note:** If the node's *Mode* is **MinManaged**, then the **makenode** command is not run.
10. When the *Mode* attribute for this node is changed to **Managed**, the node is installed and ready for use in the cluster. For **MinManaged** nodes, the *Mode* will not be changed to **Managed**. Run the **monitorinstall** command from the management server to see the installation status of **MinManaged** nodes.

During an **AutoYaST** install, the **installnode** command completes the following steps:

1. Sets up the **/tftpboot/pxelinux.cfg/node-ip-addr-in-hex** file for each node, so that when a node reboots it does an AutoYaST installation.
2. Reboots the node.



3. As the node reboots, broadcasts its MAC address.
  4. The **dhcp** server on the management server accepts the node's **dhcp** request, and **pxelinux** is used to initiate an **AutoYaST** install. For Linux nodes on pSeries servers, an **arp** entry is made for the node on the management server and the node is rebooted to start in firmware mode. The **installnode** command communicates with the node's firmware to retrieve the kernel and initial directory and to boot.
  5. AutoYaST installs the operating system on each node.
  6. After the AutoYaST installation of the operating system is complete, the AutoYaST post-installation script **/csminstall/csm/InstallCSMVersion/autoyast. \ InstallDistributionNameInstallDistributionVersion/nodeipaddr-autoyast.xml** is run on the node. After some additional configuration, any user scripts in **/csminstall/csm/scripts/installprereboot** are run. Then, the AutoYaST post-install script adds the **csmfirfirstboot** script, **/opt/csm/install/csmfirfirstboot**, to **/etc/inittab**. It then modifies the **pxelinux** configuration file on the management server **/tftboot/pxelinux.cfg/node-ip-addr-in-hex** so that the next reboot of the node reboots from the local hard drive.
  7. The node reboots to the local hard drive.
  8. As the reboot completes, the **csmfirfirstboot** script, which was listed in **/etc/inittab**, is run on the node.
  9. The **csmfirfirstboot** script runs **makenode** and any user scripts in **/csminstall/csm/scripts/installpostreboot**. The **makenode** script installs CSM and its dependencies and sets its management server. Then **csmfirfirstboot** removes itself from **/etc/inittab**.
- Note:** If the node's *Mode* is **MinManaged**, then the **makenode** script is not run.
10. When the *Mode* attribute for this node is changed to **Managed**, the node is installed and ready for use in the cluster. For **MinManaged** nodes, the *Mode* will not be changed to **Managed**. Run the **monitorinstall** command from the management server to see the installation status of **MinManaged** nodes.

If the **addlcsext** command has been run on the CSM management server, the **installnode** command ensures that it is automatically run on all specified nodes.

## Options

- a** Installs all Managed nodes with the *InstallMethod* attribute set to **kickstart** or **autoyast**. This option cannot be used with the **-P** or **-N** options, or with the *node\_list* parameter.
  - file filename**  
Specifies a file containing a list of nodes names. If the *filename* specified is **-**, then the list is read from standard input. The file can contain multiple lines and each line can have one or node names, separated by spaces.
  - h** Displays command usage information.
  - n node\_list**  
Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.
- Note that you cannot use the **-n** option with the **-a**, **-P** options. Note also that the **-n** option is optional.

## installnode

- N** *nodegroups*  
Specifies a comma-separated list of node groups to install. This option cannot be used with the **-a** or **-P** options.
- P** Installs all nodes whose *Mode* attribute is set to **PreManaged** and *InstallMethod* attribute is set to **kickstart** or **autoyast**. This option cannot be used with the **-a** or **-N** options or with the *node\_list* parameter.
- v | -V**  
Writes command's verbose messages to standard output.

## Environment Variables

### CSM\_FANOUT

Sets the maximum number of concurrent reboots. If this variable is not set, 16 nodes are rebooted concurrently. If it is set to 0, all nodes are rebooted concurrently.

### CSM\_FANOUT\_DELAY

Sets the delay in seconds between rebooting groups of nodes. If this variable is not set, the delay is 1200 seconds (20 minutes).

## Security

The command requires root access to the cluster management server.

## Examples

1. To install the operating system and CSM on Managed nodes in the cluster with *InstallMethod* attributes set to **kickstart** or **autoyast**, enter:  

```
installnode -a
```
2. To install all PreManaged nodes in the cluster whose *Mode* is **PreManaged**, enter:  

```
installnode -P
```
3. To install a list of nodes, enter:  

```
installnode clsn05 clsn06 clsn14 clsn16
```
4. Install all the nodes in the node group **mynodes**, enter:  

```
installnode -N mynodes
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

### **/csm/install/csm/status/node-hostname**

Location of the installation status file used by **monitorinstall**.

### **/opt/csm/bin/installnode**

Location of the **installnode** command.

### **/var/log/csm/installnode.log**

Location of the log file on the management server for the **installnode** command. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4.

**/var/log/csm/install.log**

Location of the log file on each node containing installation information. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4.

**Location**

**/opt/csm/bin/installnode**

**See Also**

- The **addlcsext**, **csmsetupyast**, **csmsetupks**, **definnode**, and **monitorinstall** commands.
- The **kscfg.tmpl** file.
- The **yastcfg.InstallDistributionNameInstallDistributionVersion-InstallArchitecture.xml** file.
- The *IBM CSM for Linux: Planning and Installation Guide* and the *IBM CSM for Linux: Hardware Control Guide*.

**Author**

IBM CSM Development - cluster@us.ibm.com

---

## kscfg.tpl File

### Purpose

**kscfg.tpl** - Linux operating system configuration file template used by the **csmsetupks** command.

### Description

The **kscfg.tpl** file is the template used by the **csmsetupks** command to create a Kickstart configuration file for each Linux node. The template is located in */opt/csm/install/kscfg.tpl.InstallDistributionNameInstallDistributionVersion*.

**Note:** This file is not supported for Linux on pSeries clusters.

The Kickstart configuration file generated by **csmsetupks** from the template contains configuration information gathered by Kickstart during installation of the Linux operating system. The **kscfg.tpl** file can be used as is or modified. See the sample template in the *IBM CSM for Linux: Planning and Installation Guide* Appendix for instructions on how to properly modify the template. You can create your own Kickstart template file by copying the default template file and modifying it. Then, either set each node's *InstallTemplate* attribute to the file name of the template and run **csmsetupks**, or run **csmsetupks -k** to specify the template to use.

To change the settings of all nodes in the Linux cluster, modify the Kickstart configuration file template before running the **csmsetupks** command. To change node-specific settings, modify the generated Kickstart configuration file for a specific node. The generated node-specific configuration files are called */csminstall/csm/InstallCSMVersion/kickstart.\InstallDistributionNameInstallDistributionVersion/node-ipaddr-kickstart*. Modify the generated node-specific configuration file after running the **csmsetupks** command.

There are variables in the format #VARIABLE# that must not be deleted. These variables are automatically customized with the appropriate values during the process of generating the Kickstart configuration file. For example, the following are some of the variables that are automatically customized:

**#MGMTSVR\_HOSTNAME#**

Replaced with the host name of the management server.

**#NODE\_HOSTNAME#**

Filled in with the host name of the node. This is the host name by which the management server knows the node.

**#NFS\_HOSTNAME#**

Replaced with the host name of the management server.

**#NFS\_DIR#**

Replaced with the directory on the management server that contains the Red Hat installation images; for example, */csminstall/Linux/RedHat/8.0/i386*.

**#TIMEZONE#**

Filled in with the time zone of the management server (in */etc/sysconfig/clock*).

**#TIMEZONE\_UTC#**

Filled in with the time zone UTC option of the management server (in */etc/sysconfig/clock*).

The *Gateway* and *Nameserver* values are represented by attributes that can be specified on the **csmsetupks** command line. The default values for these attributes are the same as the values for these attributes on the management server.

If you modify the **kscfg.tmpl** file or a generated Kickstart configuration file for a node, do not alter sections surrounded by the following text:

```
##### DO NOT ERASE THIS SECTION (begin)#####
##### DO NOT ERASE THIS SECTION (end) #####
```

## Security

Using this file requires root access to the cluster management server.

## Implementation Specifics

This file is part of IBM Cluster Systems Management (CSM) software.

## Location

*/opt/csm/install/kscfg.tmpl.RedHat.Version*

## Files

*/csminstall/csm/InstallCSMVersion/kickstart.\*

*InstallDistributionNameInstallDistributionVersion/node-ipaddr-kickstart*

Location of node-specific Kickstart configuration file, for example:

*/csminstall/csm/1.2.0/kickstart.RedHat8.0/9.117.241.32-kickstart*

## See Also

- The **csmsetupks**, **definenode**, **rconsole**, and **rpower** commands.
- The *IBM CSM for Linux: Planning and Installation Guide* and *IBM CSM for Linux: Hardware Control Guide*.
- *The Official Red Hat Customization Guide* for the Kickstart file format.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## Ishwinfo Command

### Purpose

**Ishwinfo** - Collects node information from one or more hardware control points.

### Synopsis

```
Ishwinfo [-h] | [-v] [-o filename] -c hcp [hcp,...] -p PowerMethod
```

### Description

The **Ishwinfo** command runs on the management server and lists hardware information for each node managed by the specified hardware control points (*hcp*). The **Ishwinfo** command can only be used on nodes for which **xseries** (xSeries and BladeCenter), **hmc** (HMC-attached pSeries), or **csp** (SP Nodes and p660 servers) is the power method.

When using the **Ishwinfo** command to generate output for xSeries and BladeCenter hardware control points, note that the first line of the output always contains information about the hardware control point itself. For example, if you run the **Ishwinfo** command to determine the nodes that are connected to hardware control point 172.22.10.1, as follows:

```
Ishwinfo -p xseries -c 172.22.10.1
```

you would see output similar to the following:

```
Hostname::PowerMethod::HWControlPoint::NodeId  
no_hostname::xseries::172.22.10.1::MgmtModule:::  
blade01.clusters.com::xseries::172.22.10.1::blade01:::  
blade02.clusters.com::xseries::172.22.10.1::blade02:::  
blade03.clusters.com::xseries::172.22.10.1::blade03:::
```

In the output above, the line `no_hostname::xseries::172.22.10.1::MgmtModule:::` represents the hardware control point. The nodes connected to the hardware control point follow this line: `blade01`, `blade02`, and `blade03`.

#### Notes:

1. `no_hostname` is returned for nodes or hardware control points that have not been defined.
2. `LParId::HWType::HWModel::HWSerialNum` are returned for nodes on pSeries hardware only.
3. You can redirect the output of the **Ishwinfo** command to a file that you can then supply to the **definnode** command as a host mapping file.
4. This command is not currently supported for nodes on IntelliStation workstations.

### Parameters

None.

## Options

- c** *hcp, [hcp,...]*  
Specifies one or more comma-separated hardware control point host names or IP addresses on which to run the command. At least one hardware control point must be specified.
- h** Displays usage information.
- o** *filename*  
Writes command output to the specified file.
- p** *PowerMethod*  
Specifies the power method to use to contact the hardware control point: **hmc**, **xseries**, or **apc**.
- v** Turns verbose mode on. When you specify this option, tracing is turned on for the duration of the command. After the command is run, tracing is returned to its previous state.

## Security

The command requires a user ID with read and write access to the IBM.HwCtrlPoint resource class in the RMC **ctrmc.acls** ACL file.

This command could require a **systemid** file. For more information, see the **systemid** command.

## Examples

1. To write the nodes associated with hardware control point *frame10* and power method *xseries* to the file **/tmp/frame10\_nodes**, enter:

```
lshwinfo -p xseries -c frame10 -o /tmp/frame10_nodes
```

The output file **/tmp/frame10\_nodes** is similar to:

```
# Hostname::PowerMethod::HWControlPoint::HWControlNodeId
no_hostname::xseries::frame10.acme.com::f10n01:::
no_hostname::xseries::frame10.acme.com::f10n02:::
no_hostname::xseries::frame10.acme.com::f10n03:::
no_hostname::xseries::frame10.acme.com::f10n04:::
```

2. To write the nodes associated with hardware control points *frame10*, *frame11*, and *frame13*, and power method *hmc* to the file **/tmp/hcp\_out**, enter:

```
lshwinfo -p hmc -c frame10,frame11,frame13 -o /tmp/hcp_out
```

The output file **/tmp/hcp\_out** is similar to:

```
# Hostname::PowerMethod::HWControlPoint::HWControlNodeId::LParId:\
:HWType::HWModel::HWSerialNum
no_hostname::hmc::frame10.acme.com::node01::1::7040::670::1234567
no_hostname::hmc::frame10.acme.com::node02::2::7040::670::1234567
no_hostname::hmc::frame10.acme.com::node03::3::7040::670::1234567
no_hostname::hmc::frame10.acme.com::node04::4::7040::670::1234567
no_hostname::hmc::frame11.acme.com::node01::1::7040::670::7843245
no_hostname::hmc::frame11.acme.com::node02::2::7040::670::7843245
no_hostname::hmc::frame11.acme.com::node03::3::7040::670::7843245
no_hostname::hmc::frame11.acme.com::node04::4::7040::670::7843245
no_hostname::hmc::frame13.acme.com::node01::1::7040::670::5432545
no_hostname::hmc::frame13.acme.com::node02::2::7040::670::5432545
no_hostname::hmc::frame13.acme.com::node03::3::7040::670::5432545
no_hostname::hmc::frame13.acme.com::node04::4::7040::670::5432545
```

## Ishwinfo

```
|
| 3. The following output lists the nodes with an hmc power method and a CEC in
| LPAR (partitioned) mode but with no LPARs defined:
| # Hostname::PowerMethod::HWControlPoint::HWControlNodeId
| #no_hostname::hmc::frame10.acme.com:::::7040::670::1234567
|
| The # character indicates that there are no LPARs defined on this particular
| CEC. Also, the HWControlNodeId and LParID fields are blank.
```

## Exit Status

- 0 The command has run successfully.
- 1 An error has occurred.

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

See the output files in the Examples section.

## Location

**/opt/csm/bin/Ishwinfo**

## See Also

The **definnode** and **Ishwstat** commands.

## Author

IBM CSM Development - cluster@us.ibm.com



---

## Ishwstat Command

### Purpose

**Ishwstat** - Collects environmental and Vital Product Data (VPD) information from nodes on xSeries and BladeCenter servers.

### Synopsis

```
Ishwstat [-h] | [-v] {-a | -n node_list | -N nodegroups} {cputemp | disktemp | ambtemp | temp | voltage | fanspeed | power | powertime | reboots | state | cpuspeed | maxdimm | insdimm | memory | model | serialnum | asset | all}
```

### Description

The **Ishwstat** command runs on the management server to collect environmental and VPD information for the specified nodes on xSeries and BladeCenter servers. The command uses the CSM database to determine the nodes and their service processor information. The host name used must be the name specified when defining the nodes in the CSM database. The command prints a response for each node, specifying the node name and the result of the query.

**Note:** This command is not supported for Linux on pSeries clusters.

Note that the **Ishwstat** command can only be used on nodes for which **xseries** is the power method.

**Note:** This command does not currently support nodes on IntelliStation workstations.

### Options

<b>-a</b>	Runs the command on all nodes on xSeries and BladeCenter servers in the cluster. Note that you cannot use the <b>-a</b> option with the <b>-n</b> option.
<b>all</b>	Returns information for all parameters.
<b>ambtemp</b>	Returns the node ambient temperature.
<b>asset</b>	Returns the service processor's asset tag.
<b>cpuspeed</b>	Returns the speed of the installed CPUs in MHz.
<b>cputemp</b>	Returns current CPU temperatures.
<b>disktemp</b>	Returns current DASD temperatures.
<b>fanspeed</b>	Returns the percentage of maximum speed that the fan is rotating.
<b>-h</b>	Writes usage information to standard output.
<b>insdimm</b>	Returns the number of DIMMs installed.
<b>maxdimm</b>	Returns the maximum number of DIMMs the node can support.
<b>memory</b>	Returns the total memory installed in the node.
<b>model</b>	Returns the system model type.
<b>-n <i>node_list</i></b>	Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command.

## lshwstat

Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

Note that you cannot use the **-n** option with the **-a** option.

<b>-N</b> <i>nodegroups</i>	Specifies one or more node groups on which to run the command.
<b>power</b>	Returns the current power status.
<b>powertime</b>	Returns the number of hours the system has been running since the last restart.
<b>reboots</b>	Returns the number of times the system has been restarted.
<b>serialnum</b>	Returns the model serial number of the node.
<b>state</b>	Returns the current node state.
<b>temp</b>	Returns all temperatures: <b>cputemp</b> , <b>disktemp</b> , <b>ambtemp</b> .
<b>-v</b>	Turns verbose mode on. When you specify this option, tracing is turned on for the duration of the command. After the command is run, tracing is returned to its previous state.
<b>voltage</b>	Returns VRM, power supply, and system board voltages.

## Exit Status

<b>0</b>	The command has run successfully.
<b>1</b>	An error has occurred.

## Security

The command requires a user ID with read and write access to the IBM.NodeHwCtrl resource class in the RMC **ctrmc.acls** ACL file.

This command could require a **systemid** file. For more information, see the **systemid** command.

## Examples

1. To determine the CPU temperature on the host clsn01, enter:

```
lshwstat -n clsn01 cputemp
```

Output is similar to the following example:

```
clsn01: CPU 1 Temperature = 35.0 C (95.0 F)
CPU 2 Temperature not available.
hard shutdown: 95.0 C (203.0 F)
soft shutdown: 90.0 C (194.0 F)
warning: 85.0 C (185.0 F)
warning reset: 78.0 C (172.4 F)
```

2. To determine the DASD temperature on the host clsn01, enter:

```
lshwstat -n clsn01 disktemp
```

Output is similar to the following example:

```
clsn01: DASD 1 Temperature = 22.0 C (71.6 F)
DASD 2 Temperature not available
DASD 3 Temperature not available
```

3. To determine the voltage on host clsn01, enter:

```
lshwstat -n clsn01 voltage
```

Output is similar to the following example:

```
clsn01: System Board 5V: 5.19  
System Board 3V: 3.37  
System Board 12V: 12.4  
System Board -12V: -12.07  
System Board 2.5V: 2.51  
VRM1: 1.5  
VRM2: 0.47
```

4. To determine the CPU speed on host clsn01, enter:

```
lshwstat -n clsn01 cpuspeed
```

Output is similar to the following example:

```
clsn01: CPU Speed = 850 MHz
```

5. To determine the serial number of host clsn01, enter:

```
lshwstat -n clsn01 serialnum
```

Output is similar to the following example:

```
clsn01: Serial Number: 23A6215
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

See the output files in the Examples section.

## Location

```
/opt/csm/bin/lshwstat
```

## See Also

The **csmstat** and **lshwinfo** commands.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## Isnode Command

### Purpose

**Isnode** - Lists the node definitions in the CSM database.

### Synopsis

```
Isnode [-i | -s | -p | -a attr,... | -A] [-l | -F | -d delim | -D delim] [-x] [-S] [-H]
[-h][-v | -V] [-w selectstr | -n node_list] [-N nodegroups] [-f file]
```

### Description

The **Isnode** command lists attributes for nodes in the CSM cluster. If no nodes are specified using **-w**, **-n**, **-f**, or **-N**, then the **Isnode** command will list all the nodes in the cluster. For a complete list of attributes that the **Isnode** command can display, see the **nodeattributes** command.

If **-w *selectstr*** is specified, the **Isnode** command uses the string in the "where" specification of an SQL select statement against the database of nodes, and displays information about the nodes that are matched.

### Parameters

None.

### Options

- a *attr*** Displays the specified attribute. Attributes must be separated by commas.
- A** Displays all attributes.
- d *delim***  
This option is identical to the **-D *delim*** option.
- D *delim***  
Specifies the delimiter used to separate items within a row. The default is a comma-space. Rows should be separated by blank lines.
- f *file*** Specifies a file that contains a list of nodes. If the file name is **-**, then the list is read from standard input. The file can contain multiple lines and each line can have one or more node names, separated by spaces, commas, or both.
- F** Displays output in a format suitable for redirecting into a **nodedef** file, which can then be used as input to the **definenode -f** command. This option can be used to save the definition of one or more nodes so that they can be restored at a later time.
- h** Writes the command's usage statement to standard output.
- H** Shows attributes from the **IBM.Host** resource class on each node, instead of attributes from the **IBM.ManagedNode** resource class. The attributes from **IBM.Host** represent the actual values currently on the node. In some cases, the corresponding attributes in **IBM.ManagedNode** represent the value requested by the administrator.
- i** Displays IP addresses.
- l** Displays output in long format. The **-x** option has no effect when this option is specified.

**-n** *node\_list*

Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

**-N** *nodegroups*

Specifies a comma or space-separated list of node groups. All nodes in the specified node groups are displayed. Node groups are evaluated by the **nodegrp** command.

**-p** Displays the status of the nodes.

**-s** Displays the short host name.

**-S** Sorts the output by the first attribute displayed in each row. This option implies **-x** and cannot be used with **-i**, **-s** or **-l**. To determine which node the attributes apply to, include the *Hostname* attribute in the list.

**-v** | **-V**

Writes the command's verbose messages to standard output.

**-w** *selectstr*

Displays the nodes that match the "where" specification of the select string. Include the entire string within double quotation marks when single quotation marks are used to specify attribute values as strings. Specifying a where string of \* indicates that all nodes should be listed, as if a *where* string was not specified.

**-x** Specifies to not display the name of the node at the beginning of each row that is returned. The **-x** option has no effect when the **-l** option is specified.

## Exit Status

- 1** A combination of options and arguments entered is not valid.
- 12** Node or node group not found.
- 13** The node list file specified could not be opened.

If an error occurs in the RMC layer, the RMC return code will be returned as the exit status.

## Examples

1. To list the names of all nodes, enter:  
lsnode
2. To list the *Status* of all nodes, enter:  
lsnode -p
3. To list all of the attributes for node **websvr**, enter:  
lsnode -l websvr
4. To list all nodes whose host name starts with **clsn0**, enter:  
lsnode -w "Hostname like 'clsn0%'"
5. To list the version of the operating system that is installed on all nodes in the range clsn05 through clsn22, enter:  
lsnode -a InstallDistributionVersion clsn05-clsn22
6. To sort node output by *HWControlPoint* attribute value, enter:

## lsnode

```
| lsnode -S -a HWControlPoint,Hostname
```

```
| Output is similar to:
```

```
| hwctr11,node2  
| hwctr11,node4  
| hwctr11,node7  
| hwctr12,node1  
| hwctr12,node8
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

`/opt/csm/bin/lsnode`

## See Also

The **chnode**, **definenode**, **nodedef**, **nodegrp**, and **rmnode** commands.

The **nodeattributes** man page.

## Author

IBM CSM Development - cluster@us.ibm.com

## Issnmp Command

### Purpose

**Issnmp** - Collects SNMP agent configuration information from nodes on xSeries and BladeCenter servers.

### Synopsis

**Issnmp** [-h] | [-v] {-a | -n *node\_list* | -N *nodegroups*} {**address** | **agent** | **contact** | **events** | **location** | **community** | **traps** | **all**}

### Description

The **Issnmp** command runs on the management server to collect SNMP agent configuration information for the specified nodes on xSeries and BladeCenter servers. The command uses the CSM database to determine the nodes and their service processor information. The host name used must be the name specified when defining the nodes in the CSM database. The command prints a response for each node, specifying the node name and the result of the query.

**Note:** This command is not supported for Linux on pSeries clusters.

Note that the **Issnmp** command can only be used on nodes for which **xseries** is the power method. The command currently does not support IntelliStation workstations.

### Options

<b>-a</b>	Runs the command on all nodes on xSeries and BladeCenter servers in the cluster. Note that you cannot use the <b>-a</b> option with the <b>-n</b> option.
<b>address</b>	Returns all community addresses.
<b>agent</b>	Returns SNMP agent state.
<b>all</b>	Returns information for all parameters.
<b>community</b>	Returns all community names.
<b>contact</b>	Returns system contact name.
<b>events</b>	Returns all enabled events.
<b>-h</b>	Writes usage information to standard output.
<b>location</b>	Returns system location string.
<b>-n <i>node_list</i></b>	Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the <b>noderange</b> man page.  Note that you cannot use the <b>-n</b> option with the <b>-a</b> option.
<b>-N <i>nodegroups</i></b>	Specifies one or more node groups on which to run the command.
<b>traps</b>	Returns SNMP traps state.
<b>-v</b>	Turns verbose mode on. When you specify this option, tracing is

## lssnmp

turned on for the duration of the command. After the command is run, tracing is returned to its previous state.

## Exit Status

- 0 The command has run successfully.
- 1 An error has occurred.

## Security

The command requires a user ID with read and write access to the IBM.NodeHwCtrl resource class in the RMC **ctrmc.acls** ACL file.

This command may require a **systemid** file. For more information, see the **systemid** command.

## Examples

1. To determine if the SNMP agent on hosts clsn01 and clsn02 is enabled or disabled, enter:

```
lssnmp -n clsn01,clsn02 agent_enabled
```

Output is similar to the following example:

```
clsn01 Agent is enabled  
clsn02 Agent is disabled
```

2. To determine the SNMP community names on host clsn01, enter:

```
lssnmp -n clsn01 community
```

Output is similar to the following example:

```
clsn01  
Community 1 = Community1
```

3. To determine the SNMP event triggers enabled on host clsn01, enter:

```
lssnmp -n clsn01 events
```

Output is similar to the following example:

```
clsn01 Events Enabled:  
CRITICAL_HARD_DRIVE  
CRITICAL_MULTIPLE_FAN  
CRITICAL_POWER_FAILURE  
CRITICAL_TAMPER  
CRITICAL_TEMPERATURE  
CRITICAL_VOLTAGE  
CRITICAL_VRM_FAILURE  
NONCRITICAL_ERROR_LOG_75FULL  
NONCRITICAL_ERROR_LOG_FULL  
NONCRITICAL_RPS  
NONCRITICAL_SECONDARY
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

**/opt/csm/bin/lssnmp**



## See Also

The **chsnmp** command.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## mgmtsvr Command

### Purpose

**mgmtsvr** – Displays or sets the management server entry for the Managed node.

### Synopsis

**mgmtsvr** [-h] [-k] [-v | -V] [-N] [-n *hostname*] [**host** | -d]

### Description

The **mgmtsvr** command runs on Managed nodes to display or set the management server for the node.

### Parameters

None.

### Options

- d** Deletes the management server entry.
- h** Writes the command's usage statement to standard output.
- host** Specifies the host name or IP address of the CSM management server that will be managing this node. If no host is specified, and the **-d** option is not specified, the current setting will be displayed. In this case, the exit status can be examined to distinguish between nodes and the management server.
- k** Exchanges RSCT public keys, host name, and node identifier between the node and the management server. Only use this option if the host name, node identifier, or public key of either machines has changed. At the time this option is used, you must trust your network to ensure that a rogue machine is not going to insert its public key in place of the public key of the real node.
- n *hostname***  
Specifies the host name known to the CSM management server for this node. Passing a node name to the **updatenode** command is normally only needed for diagnostic purposes. The **updatenode** command uses the **-n** option to ensure correct operation when the *hostname* value on the node resolves to a different name than the name known to the management server. The default is **/bin/*hostname***. Before using the **updatenode** command with the **-n** option, you must run **chnode *nodename* AllowManageRequest=1**.
- N** Displays the value for the local node name and management server. This option is valid only if a host name or IP address for the management server is not specified.
- v | -V**  
Writes the command's verbose messages to standard output.

### Exit Status

- 0** This is a Managed node only and the management server value is set.
- 1** The combination of options is not valid.
- 11** This is a Managed node only and the management server value is not set.

- 21 This node is its own management server.
- 22 This node is also a management server for a separate cluster.
- 23 This machine is a CSM management server and a node, but the node does not currently have a management server set for it.
- 31 This machine is a CSM management server only.
- 101 This machine is neither a CSM management server or node.

## Security

The command requires root access to the cluster management server.

## Examples

To query the current CSM management server for this node, enter:

```
mgmtsvr
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

**/var/log/csm/install.log**

Location of the log output from the **updatenode** command.

## Location

**/opt/csm/bin/mgmtsvr**

## See Also

- The **installnode** and **updatenode** commands.
- See the cluster security section in the *IBM CSM for Linux: Administration Guide* for information on editing the RMC ACL file.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## monitorinstall Command

### Purpose

**monitorinstall** - Displays the status of the CSM installation on each of the nodes.

### Synopsis

```
monitorinstall [-h] [-v | -V] [-e] [-i] [-l] [-u][-a] [-n node_list] [-f filename] [-N nodegroups]
```

### Description

The **monitorinstall** command displays the status of CSM installation for each node. A filtering scheme can be used to display only nodes of interest. By default, all PreManaged, Managed, and MinManaged nodes defined in the cluster are displayed in the installation monitor. You can display all nodes, only installed nodes, only nodes not installed, or only nodes that did not install correctly. If no option is specified, then all nodes are displayed. To run the **monitorinstall** command at specified intervals, use the **watch** command.

### Options

- a** Specifies to monitor all nodes. This is the command default.
- n *node\_list***  
Specifies a comma or space-separated list of one or more node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotation marks. For information about specifying node ranges, see the **noderange** man page.
- N *nodegroups***  
Specifies a comma or space-separated list of one or more node groups on which to run the command. Space-separated groups must be inside double quotation marks. Status is only displayed for nodes in the specified node groups.
- e** Shows only the nodes that had an error during the installation process. This option can be used with the **-i** or **-u** option to display different types of nodes. Note that some errors are undetectable by the **monitorinstall** command. If no options are specified, then all nodes are displayed.
- f *filename***  
The name of a file containing a list of nodes to process.
- h** Writes the command's usage statement to standard output.
- i** Shows only the nodes that are installed. This option can be used with the **-e** or **-u** option to display different types of nodes. Note that some errors are undetectable by the **monitorinstall** command. If no options are specified, then all nodes are displayed.
- l** Displays output in long format.
- u** Shows only the nodes that are not installed and nodes that are in the process of installing. This option can be used with the **-e** or **-i** option to display different types of nodes. Note that some errors are undetectable by the **monitorinstall** command. If no options are specified, then all nodes are displayed.

**-v | -V**

Writes the command's verbose messages to standard output.

## Security

The command requires root access to the cluster management server.

## Examples

1. To display the progress of the installation, enter:

```
monitorinstall
```

Output is similar to the following:

Node	Mode	Status
clsn02.ppd.pok.ibm.com	Managed	Installed
clsn03.ppd.pok.ibm.com	Managed	Installed
clsn04.ppd.pok.ibm.com	MinManaged	Installed
clsn05.ppd.pok.ibm.com	MinManaged	Installed
clsn06.ppd.pok.ibm.com	MinManaged	Installed
clsn07.ppd.pok.ibm.com	Managed	Installed
clsn08.ppd.pok.ibm.com	PreManaged	Not Installed

2. To display expanded details on the progress of the installation, enter:

```
monitorinstall -l
```

Output similar to the following is displayed:

```
Node clsn02.ppd.pok.ibm.com
```

```
-----  
No status file.
```

```
Node clsn03.ppd.pok.ibm.com
```

```
-----  
Thu Oct 18 15:45:19 EDT 2001: Full Install Initiated  
Thu Oct 18 15:45:21 EDT 2001: Rebooting to Install Node
```

```
Node clsn04.ppd.pok.ibm.com
```

```
-----  
Mon Oct 22 12:40:41 EDT 2001: Full Install Initiated  
Mon Oct 22 12:40:42 EDT 2001: Rebooting to Install Node  
Mon Oct 22 08:55:24 EDT 2001: Kickstart RPM Installation Complete.  
Mon Oct 22 08:55:24 EDT 2001: Starting Kickstart Post-Install.
```

```
Node clsn05.ppd.pok.ibm.com
```

```
-----  
Mon Oct 22 12:40:41 EDT 2001: Full Install Initiated  
Mon Oct 22 12:40:42 EDT 2001: Rebooting to Install Node  
Mon Oct 22 08:55:24 EDT 2001: Kickstart RPM Installation Complete.  
Mon Oct 22 08:55:24 EDT 2001: Starting Kickstart Post-Install.  
Mon Oct 22 08:55:25 EDT 2001: Kickstart Post-Install Complete.  
Mon Oct 22 08:55:25 EDT 2001: Rebooting to hard disk.  
Mon Oct 22 08:57:41 EDT 2001: Starting Firstboot  
Mon Oct 22 08:57:41 EDT 2001: Starting makenode to install CSM  
RPMs  
Mon Oct 22 08:59:51 EDT 2001: makenode complete: status=0  
Mon Oct 22 08:59:52 EDT 2001: Firstboot complete.: status=0  
Mon Oct 22 08:59:52 EDT 2001: Installed
```

3. To continuously display the output of the **monitorinstall** command, enter:

```
watch monitorinstall
```

**monitorinstall**

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

*/csminstall/csm/status/node\_hostname*

Location of the node's install status file.

## Location

*/opt/csm/bin/monitorinstall*

## See Also

- The **installnode** and **watch** commands.
- The *IBM CSM for Linux: Planning and Installation Guide* for installation details.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## nodeattributes

### Purpose

**nodeattributes** - Displays CSM nodes attributes and values. **nodeattributes** is not a command – it is a file listing node attribute information.

### Description

The CSM node attributes and their valid values are:

#### AllowManageRequest

Allows a request to manage the node. This attribute can be edited. It is set by the **updatenode** or **csmsetupnim** command. Valid values are:

**0** – Do not allow a manage request from the node.

**1** – Allow a manage request from the node.

#### ConsoleMethod

The console method used to open a node console. This attribute can be edited. Valid values are: **hmc**, **mrsv**, **esp**, **els**, **computone** and **cps** (make sure you type these values exactly as shown here).

**Note:** CSM supports the ESP console server on Red Hat 7.2, 7.3, and 8.0 management servers only.

#### ConsolePortNum

The number associated with the console port for this node. This attribute can be edited.

The valid MRV console port numbers range from **1** through **40** depending on the model.

The valid Computone console port numbers range from **1** through **48** depending on the model.

The valid CPS and ELS console port numbers range from **1** through **16** (decimal numbers, not hexadecimal).

Valid ESP console port numbers are **0**, **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, **a**, **b**, **c**, **d**, **e**, and **f**.

**Note:** CSM supports the ESP console server on Red Hat 7.2, 7.3, and 8.0 management servers only.

This attribute is not used for the HMC.

#### ConsoleSerialDevice

The device name of the console serial port on the node. This attribute is an option when defining the resource and can be edited. It is valid only for xSeries servers. Valid values are **ttyS0**, **ttyS1**, and **NONE**. If no value is specified for this attribute when the node is created using the **definenode** command, a default value of **ttyS0** will be set. If the system has no serial port defined, or if you do not want console output redirected to the serial port, the *ConsoleSerialDevice* attribute value must be set to **NONE**. If the value is set to **NONE**, the **rconsole** command will not attempt to open a remote console session for this node. If the

## nodeattributes

*ConsoleSerialDevice* attribute is set to a null value, a device name of `ttyS1` will be assumed in order to maintain compatibility with previous releases of CSM.

### ConsoleServerName

The host name of the network interface for the console server. It is optional when defining a resource.

### ConsoleServerNumber

The number associated with the ESP console server. This attribute is required only if you are using ESP console servers. This attribute can be edited.

**Note:** CSM supports the ESP console server on Red Hat 7.2, 7.3, and 8.0 management servers only.

### CSMVersion

The version of the CSM client software currently installed on this node. This attribute is updated after installing a node or running the **updatenode** command. You should not update this attribute.

### HWControlNodeID

Identifier associated with the node power control. This attribute can be edited. This attribute must be defined to enable hardware control. If you do not set the *HWControlNodeID* attribute using the **definenode** command or a **nodedef** file, it defaults to the short host name of the node.

**Note:** For nodes with a *PowerMethod=xseries*, the *HWControlNodeID* value must be unique among all of these nodes. Otherwise, the *HWControlNodeID* value must be unique for all defined nodes with the same *HWControlPoint* attribute value.

The value is dependent on the *PowerMethod* in use, and can be one of the following:

- Text ID of the service processor for nodes on xSeries servers.
- LPAR name for nodes on IBM HMC-attached pSeries servers.
- User defined node hostname for nodes on BladeCenter servers.
- APC MasterSwitch outlet port number for nodes on IntelliStation workstations.
- If your cluster contains IBM POWER3 or 375 MHz POWER3 SMP High Nodes with expansion I/O units, use the **cspcfgio** command with the **-w** option to populate this field.
- User-defined node host name for nodes on the eServer 325.

### HWControlPoint

The host name of the network adapter for the hardware control point. This attribute can be edited. This attribute must be defined to enable hardware control.

### HWType

The hardware type. You should not update this attribute. This attribute is not valid for xSeries servers.

### HWModel

The hardware model number. You should not update this attribute. This attribute is not valid for xSeries servers.

### HWSerialNum

The hardware serial number. You should not update this attribute. This attribute is not valid for xSeries servers.

### Hostname

The host name associated with the node's network interface as



known by the management server. This is the network interface used to install and manage the node.

#### **InstallAdapterDuplex**

The duplex value of the adapter that will be used during the network installation of the node. This attribute can be edited. This attribute is not valid for xSeries servers.

#### **InstallAdapterGateway**

The IP address or host name of the network gateway used to connect the node to the boot server, if they are not on the same network segment. This value can be edited, or entered with the **getadapters -w** and **-G** options. This attribute is valid for nodes on HMC-attached pSeries servers only.

#### **InstallAdapterMacaddr**

The machine address of the network adapter used for installation. This attribute can be edited.

#### **InstallAdapterSpeed**

The adapter speed used during network installation of the node. This attribute can be edited. This attribute is not valid for xSeries servers.

#### **InstallAdapterType**

The adapter type used during network installation of the node. This attribute can be edited. This attribute is not valid for xSeries servers.

#### **InstallCSMVersion**

The CSM version to be installed on the node. This is specified as *version.release.modification*, for example:

1.3.2

The default is to use the same version of CSM as that of the management server. This attribute can be edited.

#### **InstallDisk**

Specifies the device used to install Linux and CSM; for example, **/dev/sda** or **/dev/hdb**. This attribute is only used for SuSE or SuSE SLES nodes.

#### **InstallDiskType**

The disk type to be installed. The valid values are **scsi** or **ide**. The default disk type is **scsi**. This attribute is only used for SuSE or SuSE SLES nodes.

#### **InstallDistributionName**

The distribution name to be installed. The valid values are **RedHat**, **RedHatAS**, **SuSE**, and **SLES**. The default is the distribution on the management server. This attribute can be edited.

#### **InstallDistributionVersion**

The distribution version to be installed. The format for the values is *version.release*. For example,

8.0

This attribute can be edited.

**Note:** For RedHat 9, the value should be **9**, not **9.0**.

### InstallKernelVersion

The kernel version to be installed. This attribute applies to SuSE and SuSE SLES. This attribute is not used in the current CSM release.

**InstallMethod** The installation method. This attribute is set by the **csmssetupks** or the **csmssetupyast** command. The valid values are **kickstart** and **autoyast**.

### InstallOSName

The operating system to be installed. The expected value is **Linux**. The default uses the same operating system as that on the management server. This attribute can be edited.

### InstallPkgArchitecture

The machine architecture of the node. This attribute is used to determine the package to be installed on the node. For xSeries, the value should be **i386**. For pSeries, the value should be **ppc64**. The default is to use the same architecture as that of the management server. This attribute can be edited.

**Note:** The value **i386** represents any machine whose architecture is **i386**, **i486**, **i586** or **i686**. For xSeries 335, the value should be **x86\_64**.

### InstallServer

This attribute is valid on CSM for Linux only. The machine name of the install server and an optional directory that will server the CSM install files to this node. The machine name and directory must be separated by a colon ( : ). For example, to specify only the name of the install server, *nfsserver1.clusters.com*, you would enter:

```
nfsserver1.clusters.com
```

To specify the name of the install server, *nfsserver1.clusters.com*, and a directory called */mydirectory*, you would enter the following:

```
nfsserver.clusters.com:/mydirectory
```

If you do not specify a directory for the CSM repository, **/csmserver** is used. If you do not specify an install server, the management server will be used to serve the install files.

**Note:** If the install server is also a node in the cluster, you must use the host name that is stored in the *Hostname* attribute.

### InstallTemplate

The Kickstart or AutoYAST template file used to install this node. This attribute can be edited

### LastCFMUpdateTime

The time since CFM updated the node, in seconds since the epoch. Do not edit this attribute.

### LParID

The logical partition identifier. Do not edit this attribute. This attribute is not valid for xSeries servers.

### ManagementServer

The host name of the management server machine as known by the node. The *ManagementServer* attribute is set by the **definnode** command. This attribute can be edited (Read/Write).

<b>Mode</b>	The current phase of the node's installation. The mode is automatically set by CSM installation tools to <b>PreManaged</b> , <b>Installing</b> , or <b>Managed</b> , depending on the phase of the node's installation. You can manually set the mode to <b>MinManaged</b> to indicate Minimally Managed nodes. See the <i>IBM CSM for Linux: Planning and Installation Guide</i> for details.
<b>Name</b>	The host name of the node. This attribute is <b>read</b> only.
<b>PowerMethod</b>	The power method is used to perform hardware control. The valid values for Linux are <b>hmc</b> , <b>xseries</b> , <b>apc</b> , and <b>bmc</b> . This attribute can be edited. This attribute must be defined to enable hardware control.
<b>UniversalId</b>	The unique node identifier. Do not update this attribute.
<b>UpdatenodeFailed</b>	Status of the last run of the <b>updatenode</b> command on this node. This attribute will be set to <b>0</b> if the <b>updatenode</b> command was successful. Any other value means that an error occurred. Do not update this attribute.
<b>UserComment</b>	Comments provided by the administrator about the node. This attribute can be edited.

The following describes the dynamic **resource** attributes that can be monitored for the IBM.ManagedNode class:

#### **ChangedAttributes**

Provides a list of the persistent attributes that have changed.

#### **ConfigChanged**

Generates an event when a resource attribute has changed.

#### **InstallStatus**

The current installation status of the node: **PreManaged**, **MinManaged**, **Managed**, or **Installing**.

#### **PowerStatus**

The current power status of the node. The valid states are ON (1), OFF (0), UNKNOWN (127) and UNCONFIGURED (128)

#### **Status**

The current accessibility status of the node. *Accessibility* is defined as the ability to successfully communicate with the RMC subsystem on the node. The valid states are UNREACHABLE(0), ALIVE (1) and UNKNOWN (127).

## See Also

- The **chnode**, **definnode**, and **lsnode** commands.
- The *IBM CSM for Linux: Hardware Control Guide* and *IBM CSM for Linux: Planning and Installation Guide*.

## Author

IBM CSM Development - cluster@us.ibm.com

## nodedef File

### Purpose

**nodedef** - The CSM node definition file.

### Description

The **nodedef** node definition file contains node attributes that can be used to create CSM node definitions. The file is used as input to the **definenode** command.

The **nodedef** file is structured in stanza format, where each stanza contains information for a unique node. The following rules must be followed when creating a **nodedef** file:

- A stanza header consists of the node name followed by a colon (:).
- Attribute lines must take the form of *Attribute=Value*.
- A warning message is issued for attributes that are not valid, but the code continues.
- Each line of the file can have only one header or attribute definition.
- Only one stanza can exist for each node.
- If the header keyword is **default**, the attribute values in the stanza are considered default values for subsequent node definitions.
- Default stanzas can be specified multiple times and at any point in a definition file. The values apply to all definitions following the default stanzas in a file. The default values are cumulative; a default attribute value will remain set until it is explicitly unset or changed.
- To turn off a default value, set the attribute to have no value using a blank space.
- When a specific value for an attribute is provided in the node stanza, it takes priority over any default value that had been set.
- Comments beginning with the # pound sign may be added to the file. A comment must be on a separate line.
- When parsing the file, tab characters and spaces are ignored.
- By default, some attribute values will be set by the **definenode** command when the **nodedef** file is used as input.

For information about node definition attributes, see the **nodeattributes** man page.

A **nodedef** file can be created in three ways:

- Manually create a file and type in the node information.
- Generate a **nodedef** file by running the **definenode** command with the **-s** option.
- Generate a **nodedef** file containing existing node definitions by running the **lsnode** command with the **-F** option.

### Examples

A sample node definition file is available in: **/opt/csm/install/nodedef.sample**. A sample node attributes table is available in the *IBM CSM for Linux: Hardware Control Guide*.

### Implementation Specifics

This file is part of IBM Cluster Systems Management (CSM) software.

**Location**

No required location.

**Files**

**/opt/csm/install/nodedef.sample** Location of the sample node definition file.

**See Also**

- The **definnode** and **lsnode** commands.
- The *IBM CSM for Linux: Hardware Control Guide* and the *IBM CSM for Linux: Planning and Installation Guide*.

**Author**

IBM CSM Development - cluster@us.ibm.com

---

## nodegrp Command

### Purpose

**nodegrp** - Manages node group definitions in the Cluster Systems Management (CSM) database.

### Synopsis

```
nodegrp {-h | -f filename | -L | -p | -a node_list | -n node_list | -x node_list | -D
| -I | -w selectstr | -W | -s hostname} [-g] [-v | -V] [-d delim] [group]
```

```
nodegrp [-I | -S group1,group2]
```

### Description

The **nodegrp** command lists and updates CSM node groups and can be run only on the management server. Node groups can be either explicit lists of node host names created with the **-a** option, or a dynamic group that uses a select string specified with the **-w** option. When **-w** *selectstr* is specified, the **nodegrp** command uses the select string against the attributes in the IBM.ManagedNode resource class to dynamically determine the group membership.

Options can also be used to list node groups, delete a node group, remove nodes from a node group, and add an existing node group to a group. The group specified as input to the command will be affected by all options specified with the command. The *host* parameter can be specified by either a host name or an IP address. The *group* parameter is the name of the node group to be acted on.

### Parameters

*group* Specifies a node group on which to run the command. You can specify multiple groups with the **-p**, **-L**, or **-D** options, or no options .

### Options

- a** *node\_list* Adds host names to a group, and creates a group if necessary. A node list consists of one or more comma-separated host names. To specify an existing node group to add to the group, use the + sign before the group name. Note that you can also specify node ranges. See the **noderange** man page.
- d** *delim* Specifies the delimiter used to separate the items returned.
- D** Deletes a node group.
- f** *file* Specifies a file that contains a list of groups to be defined. If the file name specified is -, then the list is read from standard input. Each line of the file should contain the group name followed by a colon, then the following three fields separated by commas: type (**static** or **dynamic**), the validation setting (**validated** or **not validated**), and the definition of the group contents (either a select string or the unexpanded member list). This is same format that is output by the **-L** option.
- g** Specifies that the names in the group represent generic names that do not have to be actual node names in the CSM cluster. This option can be used only with **-a** or **-n** when creating a group.
- h** Writes the command's usage statement to standard output.

- I** (uppercase **i**) Specifies to take the intersection of two groups and display the results. Neither group is modified.
- l** (lowercase **L**) Lists all defined node groups.
- L** Lists the definitions of all node groups. Each line of output represents one group and displays its name, the type (**static** or **dynamic**), the validation setting (**-g**), and the definition of the group contents (either a select string or the unexpanded member list).
- n *node\_list*** Replaces the current group membership with the given list of nodes, creating the group if it does not exist yet. The same values for *node\_list* are supported as for the **-a** option. Note that you can also specify node ranges. See the **noderange** man page.
- p** Displays the expanded group member list. If no option is specified, this is the default.
- s *hostname*** Displays all the static and dynamic group names that contain the node represented by the specified host name or IP address.
- S** Subtracts the membership of the second group from the first group and displays the resulting members. Neither group is modified.
- v | V** Writes the command's verbose messages to standard output.
- w *selectstr*** Specifies the "where" clause of a select string for searching the IBM.ManagedNode resource class to dynamically determine the list of nodes in the group. The group is created if it does not already exist.
- W** Displays the "where" clause of the select string set for this node group.
- x *node\_list*** Removes nodes from a group. The same values for *node\_list* are supported as for the **-a** option.

## Exit Status

- 1** The combination of options and arguments entered is not valid.
- 12** The group was not found.
- 13** Two groups are required for this operation.
- 14** Cannot add nodes to a group that has the "where" string set.
- 15** Cannot remove nodes from a group that has the "where" string set.
- 16** Cannot set the "where" string on a group that has a node list.
- 17** Cannot query the "where" string on a group that has a node list.
- 18** An error occurred opening the file to read group definitions.
- 19** The format in a group file is incorrect.

## Security

The command requires root access to the cluster management server.

## Examples

1. To list all node groups in the cluster, enter:  
nodegrp

## nodegrp

2. To create a node group called **g1** that contains nodes **websvr1** and **websvr2**, enter:  

```
nodegrp -a websvr1,websvr2 g1
```
3. To list the members of the node group **g1**, enter:  

```
nodegrp g1
```
4. To add a node **websvr3** to node group **g1**, enter:  

```
nodegrp -a websvr3 g1
```
5. To remove **websvr2** from node group **g1**, enter:  

```
nodegrp -x websvr2 g1
```
6. To create a node group **ws** containing all nodes starting with **websvr**, enter:  

```
nodegrp -w "Hostname like 'websvr%'" ws
```
7. To create a node group called **total** that contains node group **g1** and node **websvr4**, enter:  

```
nodegrp -a +g1,websvr4 total
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

`/opt/csm/bin/nodegrp`

## See Also

The **chnode**, **lsnode**, **predefined-nodegroups**, and **rmnode** commands.

## Author

IBM CSM Development - cluster@us.ibm.com



---

## noderange

### Purpose

**noderange** – Node ranges provide a short hand way to specify a list of nodes to CSM commands.

### Synopsis

```

1 nodeNamePrefixAnd#-nodeNamePrefixAnd#
2 nodeNamePrefixAnd##+
3 #-#
4 #+#
5 +nodeGroup
6 +groupPrefixAnd#-groupPrefixAnd#
7 +groupPrefixAnd#-groupPrefixAnd#
8 -nodeRange

```

### Description

Node ranges are a short hand way to specify lists of nodes. You can use them with most CSM commands that accept nodes as arguments (for example, **lsnode**). See below for a list of the commands that support node ranges. You can specify multiple node names and node ranges together, separated by spaces or commas. However, within each node range, no spaces are allowed. If multiple node names and node ranges are separated by spaces and passed to an option, like **-n**, the whole list must be enclosed by quotes. For example:

```
lsnode -n "n1 n3-n9"
```

The following sections explain several ways to specify node ranges. The numbers of the sections below correspond to the numbered lines of the syntax, above.

- (1) Two node names, separated by a minus sign. If the node names have a common base and end with an integer, the range expands to the first and second nodes that are specified, plus all the nodes in between. If the string does not conform to this pattern, the entire string is treated as a node name. This means that most node names that contain dashes are still treated as regular host names.

In the syntax above, this is shown by line 1 (*nodeNamePrefixAnd#-nodeNamePrefixAnd#*). For example: *node1-node3*.

Nodes may also be specified as fully-qualified host names. For example:

```
node1.cluster.com-node3.cluster.com
```

The example above would expand to *node1.cluster.com,node2.cluster.com,node3.cluster.com*. Note that the `CSM_NODE_PREFIX` environment variable is not applied to nodes that are specified as fully-qualified host names. `CSM_NODE_PREFIX` is only applied to node names that only contain integers.

- (2) One node name with a plus sign followed by an integer. This range starts with the given node name and increments its number *n* times, where *n* is the integer that is specified after the plus sign.

In the syntax above, this is shown by line 2 (*nodeNamePrefixAnd##+*). For example, *node5+10*. In this example, the node range would include eleven nodes altogether (node5 through, and including, node15).

Nodes can also be specified as fully-qualified host names. For example:

```
node5.cluster.com+10
```

## noderange

Note that the `CSM_NODE_PREFIX` environment variable is not applied to nodes that are specified as fully-qualified host names. `CSM_NODE_PREFIX` is only applied to node names that only contain integers.

- (3, 4) For either of the two forms above, if the node names are only integers, they are prefixed by the value of the environment variable `CSM_NODE_PREFIX`, if it is set. Also, if a node name that is specified by itself (in other words, not a node range) is only an integer (it contains no alphabetic or special characters), then the node name will be prefixed by the value of the `CSM_NODE_PREFIX` environment variable, if it has been set.

In the syntax above, this is shown by line 3 (`#-#`) and line 4 (`#+#`). For example: `1-20` and `5+10`.

- (5) A name that is preceded by a plus sign is interpreted as a node group, if a node group exists with that name (without the plus sign). Otherwise, the plus sign and the name are treated as a node name.

In the syntax above, this is shown by line 5 (`+nodeGroup`). For example: `+group1`.

- (6) A plus sign followed by a node group name, a minus sign, and another node group name. This is a node group range, and is similar to a range of nodes. If the beginning and ending group names have the same base and both end in integers, the range will be expanded to include the beginning and ending groups and all the groups in between. If the group range does not conform to this format, it will be interpreted as a single group name. The group range can also be specified as a starting group name followed by a plus sign and an integer, much like a range of nodes can be specified. In the syntax above, this is shown by line 6. For example: `+group1-group4` would expand to:

**group1,group2,group3,group4**. Using the plus syntax, this range could also be specified as: `+group1+3`.

- (7) A plus sign followed by a node group name, a minus sign, and another node group name. This is a node group range, and is similar to a node range. If the beginning and ending node group names have the same base and both end in integers, the range will be expanded to include the beginning and ending groups and all the groups in between. If the group range does not conform to this format, it will be interpreted as a single group name. The group range can also be specified as a starting group name followed by a plus sign and an integer, much like node ranges can be specified. In the syntax above, this is shown by line 6. For example, `+group1-group4` would expand to **group1,group2,group3,group4**. Using the plus syntax, this range could also be specified as `+group1+3`.

- (8) When any of the above forms are preceded by a minus sign, that node range is excluded from the total list of nodes. In the syntax above, this is shown by line 7 (`-nodeRange`). For example, `-node3-node6`.

**Note:** If the first node begins with a dash (exclusion), you must provide the node range as an argument to the `-n` option.

Any node ranges that are specified to a command using the `-n` option, or as positional arguments, are combined with other options that provide ways to specify nodes (for example, `-N` and `-f`). The exclusions are processed after the total list of nodes has been accumulated.

Before using a node range in a disruptive command like `installnode` or `rmnode`, first verify that the range specifies the correct nodes by using the `lsnode` command.

## CSM commands that support node ranges

The CSM commands that support node ranges are:

- **cfmupdatenode**
- **chnode**
- **chrconsolecfg**
- **chsnmp**
- **csmsetupks**
- **csmsetupyast**
- **definenode**
- **dsh**
- **getadapters**
- **installnode**
- **lshwstat**
- **lsnode**
- **lssnmp**
- **monitorinstall**
- **netboot**
- **nodegrp**
- **rconsole**
- **reventlog**
- **rmnode**
- **rpower**
- **smsupdatenode**
- **updatenode**

The **dcem** command does not currently support node ranges

The **RSCT** commands that do not currently support node ranges are:

- **chcondition**
- **chresponse**
- **chsensor**
- **chsrc**
- **lsaudrec**
- **lscondition**
- **lsresponse**
- **lsrsrc**
- **lssensor**
- **mkcondition**
- **mksensor**
- **rmaudrec**
- **rmcondition**
- **rmresponse**
- **rmsensor**
- **rmrsrc**

### Notes:

1. The **definenode** command also supports **-n**, which specifies a starting node and **-c**, which specifies a count. The **definenode** command increments the IP address of the starting node to retrieve the additional nodes. Node ranges increment the host name.
2. When node ranges are specified with the **nodegrp** command using the **-a,-n**, or **-x** options, the node ranges are expanded into a full node list before adding them to, or removing them from, the node group. The only exception is that sub node groups, specified with **+**, are not expanded until the membership of the node group is queried.

### Differences from xCAT ranges

Some users may be familiar with the node ranges supported by xCAT. The CSM node ranges are very similar, with the following differences:

- Non-numeric ranges that are based on the definition order of nodes is not supported.
- A file listing nodes can be input with the **-f** option instead of the **^** syntax.
- Regular expressions against node names (and other attributes) can be specified using dynamic node groups, instead of the **@** syntax.
- A suffix on the node names within a node range is not supported.
- Groups are specified with the **+** operator or with the **-N** option.
- Ranges with node names that contain dashes are not supported.
- Unless the **CSM\_XCAT\_COMPAT** environment variable is specified, **n1+2** means **n1, n2, n3**, not **n1, n2**, like it does in xCAT.

## Environment Variables

### CSM\_NODE\_RANGES

If this environment variable is set to any value, it disables expansion of node ranges.

### CSM\_NODE\_PREFIX

Text to prepend to ranges that are only integers. For example, **1-5** or **1+5**.

### CSM\_XCAT\_COMPAT

If this environment variable is set to **1**, it specifies that the **+** node range syntax should behave like it does in xCAT. For example, **n1+2** would mean **n1, n2**.

## Examples

The following examples show node ranges and how CSM interprets them.

### node01-node05

node01, node02, node03, node04, node05

### n8-n11

n8, n9, n10, n11

### n1-n3,n6-n8

n1, n2, n3, n6, n7, n8

### frame2n5-frame2n7

frame2n5, frame2n6, frame2n7

**15-20** n15, n16, n17, n18, n19, n20 (assuming **CSM\_NODE\_PREFIX** is set to **n**)

**n8+3** n8, n9, n10, n11

**8+3** n8, n9, n10, n11 (assuming CSM\_NODE\_PREFIX is set to n)

**8** n8 (assuming CSM\_NODE\_PREFIX is set to n)

**+g1** n1, n2, n3 (assuming g1 is a group containing n1, n2, n3)

**n1-n3, -n2**  
n1, n3

**n1-n5, -n2-n4**  
n1, n5

**n1-n5, ++g1**  
n4, n5 (assuming g1 is a group containing n1, n2, n3)

1. To update node7 through node20, but skip node15, enter:

```
updatenode node7-node20,-node15
```

2. To list node1 and the next four nodes and node20, enter:

```
lsnode node1+4 node20
```

3. An alternate way to list the same nodes is:

```
lsnode -n node1+4,node20
```

4. To create a node group called group1, consisting of node1, the next four nodes and node20, enter:

```
nodegrp -n "node1-node4 node20" group1
```

## Implementation Specifics

This feature is part of IBM Cluster Systems Management (CSM) software.

## See Also

- The **lsnode** and **nodegrp** commands.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## predefined-condresp Command

### Purpose

**predefined-condresp** – Defines default conditions and responses for monitoring.

### Synopsis

**predefined-condresp** [-h] [-d | -f | -m | -r] [-v | -V]

### Description

The **predefined-condresp** command defines conditions and responses that are useful for monitoring. These conditions and responses are then used by the Event Response resource manager (ERRM) to monitor the cluster for the defined conditions. When a condition is detected, ERRM runs the corresponding response defined for that condition.

The **predefined-condresp** command is run automatically when the **csm.server** package is installed. The command can also be run manually to restore conditions and responses to their initial definitions. If run with no options, the **predefined-condresp** command defines only conditions and responses that do not already exist. The command options can be used only to define or remove conditions and responses.

Predefined conditions and responses for the Managed Node resource class, the Node Group resource class, the Hardware Control resource class, and the Hardware Control Point resource class are described in the *IBM CSM for Linux: Administration Guide*.

### Parameters

None.

### Options

- d** Defines the conditions and responses. This will only define conditions and responses that are not already defined. This is the default.
- f** Restores the original settings of all predefined conditions and responses. This option first removes all predefined conditions and responses and then recreates them using the original settings shipped with CSM.
- h** Displays command usage information.
- m** Specifies migration mode, which defines any new or changed conditions and responses since the last CSM release.
- r** Removes the predefined conditions and responses.
- v | -V** Writes the command's verbose messages to standard output.

### Exit Status

- 0** The command has run successfully.
- 1** The combination of options and arguments is not valid.
- n** Specifies the highest exit code returned by any of the ERRM commands.

## Security

The command requires root access to the cluster management server.

## Examples

1. To remove the current predefined conditions and responses and recreate them, enter:  
`predefined-condresp -f`
2. To define the predefined conditions and responses, enter  
`predefined-condresp -d`

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

`/opt/csm/bin/predefined-condresp`

## See Also

- The **mkcondition**, **mkcondresp**, and **mkresponse** RSCT commands.
- See the ERRM section in the *IBM CSM for Linux: Administration Guide* for information on ERRM and the predefined conditions and responses available for monitoring.

## Author

IBM CSM Development - cluster@us.ibm.com

## predefined-nodegroups Command

### Purpose

**predefined-nodegroups** – Defines predefined node groups to their initial states.

### Synopsis

**predefined-nodegroups** [-h] [-d | -f | -m | -r] [-v | -V]

### Description

The **predefined-nodegroups** command runs automatically during CSM installation, but can also be run manually to restore predefined node groups to their initial state.

### Parameters

None.

### Options

- d** Defines the node groups. This will only define node groups that are not already defined. This is the default.
- f** Restores the original settings of all predefined node groups. This option first removes all predefined node groups and then recreates them using the original settings shipped with CSM.
- h** Displays command usage information.
- m** Specifies migration mode, which defines any new or changed node groups since the last CSM release.
- r** Removes the predefined node groups.
- v | -V** Writes the command's verbose messages to standard output.

### Exit Status

- 0** The command has run successfully.
- 1** The combination of options is not valid.
- n** Displays the highest exit code returned by the **nodegrp** command.

### Security

The command requires root access to the cluster management server.

### Examples

1. To define the predefined node groups, enter:  
`predefined-nodegroups -d`
2. To remove the current predefined node groups and redefine them, enter:  
`predefined-nodegroups -f`

### Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.



**Location**

`/opt/csm/bin/predefined-nodegroups`

**See Also**

- The **nodegrp** command.

**Author**

IBM CSM Development - cluster@us.ibm.com

---

## probemgr Command

### Purpose

**probemgr** - Runs CSM cluster system diagnostics.

### Synopsis

```
probemgr [-h] [-d] [-c {0 | 10 | 20 | 127}] [-l {0 | 1 | 2 | 3 | 4}] [-e
probes_to_exclude] [-D user_probe_directory] [-a | [-n node_list] [-N nodegroups]]
[-p probe_name]
```

### Description

The **probemgr** command is part of the CSM diagnostics infrastructure, which can be used by system administrators to determine the root cause of system problems. The **probemgr** command runs a set of small diagnostics programs called *probes*. Each probe provides diagnostics for a particular part of the system. See the Diagnostics chapter of the *IBM CSM for Linux: Administration Guide* for the specific CSM probes available.

Because components of the system often depend on other parts of the system, the probes must be run in a specific order, based on how they depend on one another. Each probe creates its own dependency list and provides it to the **probemgr** command. The **probemgr** command then uses the dependency lists to build a dependency tree of probes and runs the probes based on their position in the tree. The **probemgr** command then returns the results to the system administrator.

For example, given the following dependencies:

- Probe A depends on Probe B
- Probe B depends on Probe C and Probe D

the probes would be run in the following order:

1. Probe C and Probe D (the order of C and D are not important)
2. Probe B
3. Probe A

### Parameters

None.

### Options

**-a** Runs the specified probes on all nodes in the CSM database.

**-c 0 | 10 | 20 | 127**

The highest level of exit code, returned by a probe, that the probe manager permits before terminating. The default value is 10.

**0** Success

**10** Success with Attention messages

**20** Failure

**127** Internal error

**-d** Shows the dependencies and run order of the probes.

- D** *user\_probe\_directory*  
The directory where user-specific probes reside. The default probe directory (**/opt/diagnostics/probes**) is examined after the user-specified directory is examined. If there are duplicate nodes, the probes in the user-specified directory take precedence over the probes in the default directory.
- e** *probes\_to\_exclude*  
Generates a list of probes to exclude when creating the probe dependency tree. This also means that those probes will not be run.
- h**  
Displays usage information.
- l** **0** | **1** | **2** | **3** | **4**  
Indicates the message output level. The default is 3, which indicates that Attention and Error messages must be reported by the probes.
  - 0** Specifies that the probe manager message, probe trace messages, probe explanation and suggested action messages, probe attention messages, and probe error messages should be reported.
  - 1** Specifies that the probe trace, probe explanation and suggested action messages, probe attention messages, and probe error messages should be reported.
  - 2** Specifies that the probe explanation and suggested action messages, probe attention messages, and probe error messages should be reported.
  - 3** Specifies that only the probe attention messages and probe error messages should be reported.
  - 4** Specifies that only the show probe error messages should be reported.
- n** *node\_list*  
Specifies a comma or space-separated list of host names or node ranges on which to run the specified probes. Space-separated lists must be inside double quotes. See the **noderange** man page for information on node ranges.
- N** *nodegroups*  
Specifies a comma or space-separated list of node groups on which to run the specified probes. Space-separated lists must be inside double quotes. The node groups are evaluated by the **nodegrp** command.
- p** *probe\_name*  
Specifies to run the probe indicated by *probe\_name* and all its dependencies. If this option is not specified, all probes in the default probe directory (**/opt/diagnostics/probes**) and all probes in the user probe directory (specified by the **-D** option) are run. See the Diagnostics chapter of the *IBM CSM for Linux: Administration Guide* for the specific CSM probes available.

## Environment Variables

### **STD\_PROBE\_CONFIGDIR**

The standard directory where configurations of the probes, shipped as part of the diagnostic probes package, are stored. The directory

## probemgr

name must not be relative – it must begin with a */*. If this variable is not set, the assumed location of the configurations is ***/opt/diagnostics/config***.

### **USER\_PROBE\_CONFIGDIR**

The directory where configurations of the probes, developed by individual users, are stored. The directory name must not be relative.

### **PROBE\_IGNOREFILE**

The file containing the list of probes that must not be run during the invocation of the probe manager. If other probes express dependency on these probes, the probe manager assumes that the probes that were ignored ran successfully.

Each line in the file contains a name of a probe to be ignored. The following shows the contents of a sample *ignore probes* file:

```
ProbeName1  
ProbeName2
```

If the **-e** command line option is used to specify a list of probes to ignore. The probe list to be ignored is the combination of those specified on the command line and those specified in the file.

### **PROBE\_IGNOREMSGFILE**

The file containing a list of messages to be ignored for specific probes. The probe manager reads this file and passes the messages to be ignored to the appropriate probes.

Each line in the file contains a probe name and a comma-separated list of message numbers to be ignored for that probe. The user can also specify a list of words that appear in the message, enclosed in brackets [ ] to indicate the specific message to be ignored. To identify the message to be ignored, the probes must search the message to see if it contains the message number and all the words specified by the user. The following shows the contents of a sample *ignore probe messages* file:

```
ProbeName1 MSGNUM1 [en0] ,MSGNUM2 ,MSGNUM3 [en0]  
ProbeName2 MSGNUM1 ,MSGNUM2 [/tmp 90]  
ProbeName3 MSGNUM4
```

### **STD\_PROBE\_UTILDIR**

The directory where the utilities that are shipped as part of the diagnostic probe package are stored. The directory name must not be relative – it must begin with a */*. If the variable is not set, assume that the utilities are located in ***/opt/diagnostics/utilities***.

### **USER\_PROBE\_UTILDIR**

The directory where the utilities developed by users are stored.

## Exit Status

The following diagnostics may be issued on standard error:

### **Cyclic Dependency**

Cyclic dependency between probes X and Y.

### **Probe Problems**

System error when trying to run probe X. Exiting probe manager because of probe problems.

## Security

The command requires root access to the cluster management server.

## Examples

1. To run all probes enter:  
`probemgr`
2. To show dependencies and the run order, but not run any probes, enter:  
`probemgr -d`
3. To show all messages, enter:  
`probemgr -l 0`

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

`/opt/diagnostics/bin/probemgr`

## See Also

- The diagnosis chapter of the *IBM CSM for Linux: Administration Guide*.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## rconsole Command

### Purpose

**rconsole** – Provides remote console support for CSM nodes.

### Synopsis

```
rconsole [-h] [-o | -O columns] [-a] [-t] [-c [-v] | [-r | -f]] [-n node_list] [-N nodegroups]
```

### Description

The **rconsole** command provides remote console support for nodes in a cluster. The command uses the CSM database to determine the nodes and their console access information. The **-a** option displays a console for each node defined in the cluster. The **-n** and **-N** options display a console for each node defined by these options. You can also define the **RCONSOLE\_LIST** environment variable, which is the name of a file containing a list of nodes to manage. The host names used must be the names specified when the nodes were defined in the CSM database.

The **rconsole** command uses the Conserver open source package to provide multiple read-only consoles to a node. If the **rconsole** command is run without the **-c** option, it will verify that the Conserver daemon is running before starting the console session, and will return a message if the Conserver daemon is not running. A node can have only one read-write console open at a time, but can have several read-only consoles open simultaneously. The **-r** option opens a read-only session. The **-f** option forces a read-write session to a node. If an existing read-write session is already open, it will be changed to a read-only session. The **-r** and **-f** options are mutually exclusive, and cannot be used when the **rconsole** command is run with the **-c** option.

The **rconsole** command supports the escape sequence **ctrl-E-c**, which must precede a command. When the escape sequence is entered the console application will display a left-bracket character [ and wait for a command. A command that is not valid causes the entire escape sequence to be discarded. Escape sequences cannot be used with the **-c** option. Valid escape sequence commands include:

**. (period)**

Closes the console window and disconnects.

**f** Forces a read-only session into read-write mode. Any existing read-write session is changed to a read-only session.

**o** Attempts to open (reconnect) a session.

**?** Displays the complete list of valid escape commands.

For example, to terminate a console session and exit from the **rconsole** command, enter the sequence **ctrl-E-c-period (ctrl-E-c-.)**. Use this sequence to close console windows that were not opened with the **-c** option.

The **-c** option causes the **rconsole** command to bypass Conserver and open a read-write console directly. When the **-c** option is used, the **rconsole** command does not recognize any of the preceding escape sequences, and will allow only a single read-write console session.

The **rconsole** command provides an **xterm** window for each node specified unless **-t** is specified. The font used depends on the number of nodes specified:

- 1 node**  
fixed font
- 2 - 4 nodes**  
5 x 8 font
- 5 or more nodes**  
nil2 font (not legible)

**Note:** The default **nil2** font used when five or more systems are specified is not intended to be readable; it is intended to give a general idea of whether the node is up. To read the information provided on these consoles you can define an alternate font name in the **RCONSOLE\_FONT** environment variable.

The **rconsole** command checks the *ConsoleSerialDevice* node attribute values. If the value is set to **NONE**, no remote console will be opened for that node. If the **rconsole** command is invoked for a node or list of nodes whose *ConsoleSerialDevice* attributes are all set to **NONE**, the command will return an error message. If the **rconsole** command is invoked for a list of nodes with some of their *ConsoleSerialDevice* attributes set to **NONE**, no remote console will be opened for those nodes; the remaining nodes will have remote console sessions opened. In this case, no error message will be returned.

## Options

- a** Runs the command on all of the nodes in the cluster. Note that you cannot use the **-a** option with the **-n** option. If **-a** is specified, the command ignores the **RCONSOLE\_LIST** environment variable.
- c** Bypasses Conserver and uses the console methods directly. This option cannot be used with the **-f** or **-r** options. Only one session per console is allowed when this option is used. No additional read-only sessions will be allowed.
- f** Forces a session to open in read-write mode. If a read-write session is already opened, it will be demoted to read-only. This option cannot be used with the **-c** or **-r** options.
- h** Writes usage information to standard output.
- n node\_list**  
Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.  
  
Note that you cannot use the **-n** option with the **-a** option.
- N nodegroups**  
Specifies one or more node groups on which to run the command.
- o** Displays the console windows so they do not overlap or cascade.
- O** Specifies the number of columns for the console windows.
- r** Opens the session in read-only mode. The default mode is read-write. This option cannot be used with the **-c** or **-f** options.

## rconsole

- t** Specifies text mode and displays the contents of the remote console in the current **xterm** session. This option is only allowed when one node is specified.
- v** Turns verbose mode on. Enables tracing for HMC-attached pSeries servers only. Note that the **-v** option is only valid when used with the **-c** option.

## Environment Variables

### **RCONSOLE\_FONT**

Specifies the font to use for the remote console. This overrides the default.

### **RCONSOLE\_LIST**

Specifies a file that contains a list of nodes with one host name per line.

## Exit Status

- 0** The command has run successfully.
- 1** A error occurred with one or more of the remote console commands.

## Security

For the **hmc** console method, a user ID with access to the IBM.NodeHwCtrl resource class in the RMC **ctrmc.acls** ACL file is required.

Remote console access to Linux nodes is unrestricted except for the Conserver console server. By default only root can open a remote console session to Conserver terminals. However, the **rconsole -c** bypasses Conserver, allowing a non-root user to start a console session.

## Examples

1. To open a remote console to one system, enter:  

```
rconsole -n clsn02
```
2. To open a remote console to a group of nodes defined in the CSM database as the node group **clients**, enter:  

```
rconsole -N clients
```
3. To open consoles for a group of nodes with a specific font, enter:  

```
export RCONSOLE_FONT=fixed10; rconsole -n clsn02,clsn03
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

**/opt/csm/bin/rconsole**

## See Also

- The **chrconsolecfg**, **espcfg**, **espdiag**, **esptty**, **lsnode**, and **rconsolerefresh** commands. The **espcfg**, **espdiag**, and **esptty** commands are not CSM commands – they are provided by the ESP driver software.

**Note:** CSM supports the ESP console server on Red Hat 7.2, 7.3, and 8.0 management servers only.



- The *IBM CSM for Linux: Hardware Control Guide*.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## rconsolerefresh Command

### Purpose

**rconsolerefresh** – Refreshes the Conserver daemon.

### Synopsis

**rconsolerefresh** [-h] [-r]

### Description

The **rconsolerefresh** command sends a signal to the Conserver daemon, causing it to reread its configuration file and update its internal list of consoles. If the Conserver daemon is not running, it is started. The **-r** option stops the current Conserver process and starts a new one.

Users of active console sessions can be affected in different ways by the **rconsolerefresh** command, depending on how the configuration file was changed. If a configuration file entry for an active session has been modified, running **rconsolerefresh** will cause that session to be terminated. The user will receive a message that the console session has been reset. The escape sequence **Ctrl-E-c-o** must be sent from the console session to reestablish the connection using the new configuration file entry. If multiple sessions are open for this console, the escape sequence must be given from the session that is opened for read-write.

If a configuration file entry for an active session has been removed, running the **rconsolerefresh** command will cause that session to be terminated and the console window will close without warning. For all other active sessions, running the **rconsolerefresh** command should not cause any interruption in the console session.

CSM software automatically updates the configuration file and runs the **rconsolerefresh** command whenever nodes are added to or removed from the cluster.

### Parameters

None.

### Options

- h** Writes usage information to standard output.
- r** Reinitializes the **rconsole** daemon.

### Exit Status

- 0** The command has run successfully.
- 1** An error occurred with one or more of the remote console commands.

### Security

The command requires root access to the cluster management server.

### Examples

1. To refresh or start the Conserver daemon, enter:

```
rconsolerefresh
```

2. To reinitialize the Conserver daemon, enter:

```
rconsolerefresh -r
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

```
/opt/csm/bin/rconsolerefresh
```

## See Also

- The **chrconsolecfg** and **rconsole** commands.
- The *IBM CSM for Linux: Hardware Control Guide*.
- The Conserver serial console application Web site (<http://www.conserver.com>).

## Author

IBM CSM Development - [cluster@us.ibm.com](mailto:cluster@us.ibm.com)

---

## reventlog Command

### Purpose

**reventlog** - Collects service processor log information from nodes on xSeries and BladeCenter servers.

### Synopsis

```
reventlog [-h] | [-v] {-a | -n node_list | -N nodegroups} {-e entries | clear | all | all_clear}
```

### Description

The **reventlog** command runs on the management server to collect log information for the specified nodes on xSeries and BladeCenter servers.

**Note:** This command is not supported for Linux on pSeries clusters. The command uses the CSM database to determine the nodes and their service processor information. The host name used must be the name specified when defining the nodes in the CSM database. The command prints a response for each node, specifying the node name and the result.

Note that the **reventlog** command can only be used on nodes for which **xseries** is the power method. The command is currently not supported for IntelliStation workstations.

### Options

- a**      Runs the command on all nodes in the cluster. Note that you cannot use the **-a** option with the **-n** option.
- all**      Reads all log entries, starting with the first entry.
- all\_clear**  
          Reads all log entries, starting with the first entry, and then clears the log of all entries.
- clear**    Clears the service processor log of all entries.
- e *entries***  
          Reads the number of log entries specified, starting with the first.
- h**      Writes usage information to standard output.
- n *node\_list***  
          Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.  
  
          Note that you cannot use the **-n** option with the **-a** option.
- N *nodegroups***  
          Specifies one or more comma-separated node groups on which to run the command.
- v**      Turns verbose mode on. When you specify this option, tracing is turned on for the duration of the command. After the command is run, tracing is returned to its previous state.

## Exit Status

- 0 The command has run successfully.
- 1 An error has occurred.

## Security

The command requires a user ID with read and write access to the IBM.NodeHwCtrl resource class in the RMC **ctrmc.acls** ACL file.

This command may require a **systemid** file. For more information, see “systemid Command” on page 139 or the **systemid** man page.

## Examples

1. To collect all log information on a node with host name clsn01, enter:

```
reventlog -n clsn01 all
```

Output is similar to:

```
Node01: SERVPROC Mon Mar 25 10:03:44 EST 2002 Remote Login
Successful.
Login ID: 'USERID' from CMD mode client at IP@=9.117.16.158
SERVPROC Mon Mar 25 10:00:07 EST 2002 Remote Login Successful.
Login ID: 'USERID' from CMD mode client at IP@=9.117.16.158
SERVPROC Mon Mar 25 09:53:24 EST 2002 Remote Login Successful.
Login ID: 'USERID' from CMD mode client at IP@=9.117.16.158
SERVPROC Mon Mar 25 09:52:53 EST 2002 Remote Login Successful.
Login ID: 'USERID' from CMD mode client at IP@=9.117.16.158
```

2. To clear log information on node clsn01, enter:

```
reventlog -n clsn01 clear
```

Output is similar to:

```
clsn01 clear complete rc=0
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

**/opt/csm/bin/reventlog**

## Author

IBM CSM Development - cluster@us.ibm.com

---

## rmlcsext Command

### Purpose

**rmlcsext** - Removes IBM Director Linux Cluster Support (LCS) extension components.

### Synopsis

```
rmlcsext [-h | -- help] [-A | --agent] [-S | --server] [-v | --verbose] [--restart]
```

### Description

The **rmlcsext** command removes IBM Director Server Agent and LCS extension components from IBM Director Servers, and CSM management servers and managed nodes. If you run the command without the **-A** and **-S** options, all of the found LCS extension components are removed. Either **-A** or **-S** can be specified to control which extension components are removed.

**Note:** This command is not supported for Linux on pSeries clusters.

If **addlcsext** has been run on the management server, **rmlcsext** must be run prior to running **uninstallms**.

The **rmlcsext** command is used on Linux Director Servers and on CSM management servers. On Linux Director Servers with no CSM code installed, **rmlcsext** is run directly from the CSM CD-ROM. On Windows Director Servers, **Add/Remove Programs** is used. If the **Remove** button is disabled on Windows, use the **Change** button to remove the LCS extension. On CSM management servers **rmlcsext** is run from **/opt/csm/bin**. The **rmlcsext** command is usually not run explicitly on CSM Managed nodes. If LCS extension components are installed on nodes, **rmlcsext** gets run automatically on those nodes when **uninstallms -u** is run on the management server, or when those nodes are specified by the **rmnode -u** command.

To uninstall the LCS extension, the IBM Director Server or Agent must be stopped and then restarted after the extension has been removed. Because of the potential for disruption, **rmlcsext**, if run locally without the **-- restart** option, will prompt you for permission to stop and then restart the local Server or Agent. If you decline, **rmlcsext** will not uninstall the extension and exit.

**Note:** When **uninstallms -u** or **rmnode -u** cause **rmlcsext** to run on a node, there is no prompt.

### Parameters

None.

### Options

**-A | --agent**

Removes the IBM Director Agent's LCS extension components and the **csm.director.agent** RPM package. The IBM Director Agent is not removed.

**-h | --help**

Displays command usage information.

**--restart**

Forces the IBM Director Agent or Server to restart without prompting.

**-S | --server**

Removes the IBM Director Server's LCS extension components and the **csm.director.server** RPM package. The IBM Director Server is not removed.

**-v | --verbose**

Specifies verbose mode.

## Exit Status

- 0** The command completed successfully.
- 1 - 4** The number of packages that did not uninstall.
- 96** An IBM director error occurred.
- 99** Command line arguments are incorrect.

## Security

The command requires root access to the cluster management server.

## Examples

1. To remove all LCS extension components, enter:  
`rmlcsex`
2. To remove just the IBM Director Agent's LCS extension components, enter:  
`rmlcsex -A`

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

`/opt/csm/bin/rmlcsex`

`/lcsex/rmlcsex` on CSM CD-ROM

## See Also

The **addlcsex**, **rmnode**, and **uninstallms** commands.

## Author

IBM CSM Development- cluster@us.ibm.com

---

## rmnode Command

### Purpose

**rmnode** - Removes node definitions from the CSM database.

### Synopsis

```
rmnode [-h] [-v | -V] [-u] {[-n] node_list | -f filename | -N nodegroups}
```

### Description

The **rmnode** command runs on the management server to delete Managed node definitions from the CSM database. The names of the nodes to be removed must be specified using either node host names or IP addresses. The **rmnode** command removes the node from the management server database and informs the node that it is no longer attached to the management server.

### Parameters

None.

### Options

**-h** Writes the command's usage statement to standard output.

**-f *filename***

Specifies a file that contains a list of node names. If the *filename* specified is **-**, then the list is read from standard input. The file can contain multiple lines, and each line can have one or more node names separated using commas or spaces.

**-n *node\_list***

Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

**-N *nodegroups***

Specifies a comma-separated list of node groups on which to run the command.

**-u**

Performs clean up on nodes being removed from the CSM database. The install logs and **/var/opt/csm/cfmlocal** files and directories are removed, and CSM packages are uninstalled from the nodes. This option will not uninstall any open source prerequisites or software packages that were automatically installed with the Linux operating system. If the LCS extension is installed on the node, this option also ensures that **rmlcsext** runs automatically on the nodes to remove the LCS extension.

**-v | -V**

Writes the command's verbose messages to standard output.

### Exit Status

- |    |  |
|----|--|
| 1  | A combination of options and arguments has been entered that is not valid. |
| 2  | Could not perform clean up on one or more of the specified nodes.          |
| 12 | Node not found.  |
| 13 | No nodes were specified.   |



If an error occurs in the RMC layer, the RMC return code will be returned as the exit status.

## Security

The command requires root access to the cluster management server.

## Examples

1. To remove the node named **websvr** from the CSM database, enter:  
`rmnode websvr`
2. To remove the node with the IP address 9.117.10.51 from the database, enter:  
`rmnode 9.117.10.51`

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

`/opt/csm/bin/rmnode`

## See Also

The **chnode**, **definenode**, **lsnode**, **nodegrp**, and **rmlcsext** commands.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## rpower Command

### Purpose

**rpower** – Boots, resets, powers on and off, and queries node hardware and CECs.

### Synopsis

```
rpower [-h] | [-v] {-a | -n node_list | -N nodegroups} {on | off | reboot | [-l]
query | resetsp_hcp | resetsp_host | {-m mode} cec_on | cec_off | -l
cec_query}
```

### Description

The **rpower** command requires that the **systemid** command be run first to establish the user ID and password required to access remote hardware. If you change the ID and password on the service processor, then you must rerun the **systemid** command to specify the new ID and password. See the **systemid** command for detailed information.

The **rpower** command allows for remote power control of nodes in a cluster. The command uses the CSM database to determine the nodes and their service processor information. The **-a** option causes the **rpower** command to run on all of the nodes defined in the cluster. The **-n** and **-N** options cause the command to run on the set of nodes defined by these options; you can also define an **RPOWER\_LIST** environment variable, which is a file that contains a list of nodes to manage. The host names used must be the names specified when defining the nodes in the CSM database.

The **query** option prints an **on**, **off**, **cec\_on**, **cec\_off**, and **cec\_query** status response for each of the nodes specified. The **on**, **off**, **reboot**, and **resetsp** operands print the node name with the command specified and the command return code.

### Options

**-a** Runs the command on all nodes in the cluster. Note that you cannot use the **-a** option with the **-n** option.

#### **cec\_on**

Powers on the CEC associated with the specified node. Applies to HMC-attached pSeries servers only.

#### **cec\_off**

Powers off the CEC associated with the specified node. Applies to HMC-attached pSeries servers only.

#### **cec\_query**

Returns the power status for the CEC associated with the specified node. Applies to HMC-attached pSeries servers only.

**-h** Writes usage information to standard output.

**-m** Specifies the mode to power on the CEC: **full** or **lpar**.

**-l** Returns LCD values for nodes with the **hmc** power method.

#### **-n** *node\_list*

Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated

lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

Note that you cannot use the **-n** option with the **-a** option.

- N** *nodegroups*  
Specifies one or more node groups on which to run the command.
- on** Powers the specified nodes on.
- off** Requests a shutdown and schedules a power off based on the ISP timer.
- query** Reports power status on the specified nodes.
- reboot**  
Reboots power on the specified nodes.
- resetsp\_hcp**  
Resets the service processor for the specified hardware control points.
- resetsp\_host**  
Resets the service processor on the specified nodes.
- v** Turns verbose mode on. When you specify this option, tracing is turned on for the duration of the command. After the command is run, tracing is returned to its previous state.

## Environment Variables

### RPOWER\_LIST

Specifies a file containing definitions of the hosts, one per line, which comprise the node list.

## Exit Status

- 0** The command has run successfully.
- 1** An error has occurred with one or more of the remote power commands.

## Security

The command requires a user ID with read and write access to the IBM.NodeHwCtrl resource class in the RMC **ctrmc.acls** ACL file.

This command may require a **systemid** file. For more information, see the **systemid** command.

## Examples

1. To query one node, enter:  
`rpower -n c1sn04 query`

Output is similar to:

```
c1sn04 on
```

2. To query all nodes in the cluster, enter:  
`rpower -a query`

Output is similar to:

## rpower

```
cls05.ppd.pok.ibm.com on
cls04.ppd.pok.ibm.com on
cls03.ppd.pok.ibm.com on
cls02.ppd.pok.ibm.com on
cls01.ppd.pok.ibm.com on
```

3. To query one node and have the LCD value returned, enter:

```
rpower -l -n cls05 query
```

Output is similar to:

```
cls05.pok.ibm is off
LCD1 = 0K
LCD2 =
```

4. To power off a node, enter:

```
rpower -n cls04 off
```

Output is similar to:

```
cls04 power_off complete rc=0
```

5. To display a list of the nodes in a node group, enter:

```
nodegrp -p group1
```

Output is similar to:

```
cls01.ppd.pok.ibm.com
cls02.ppd.pok.ibm.com
cls05.ppd.pok.ibm.com
```

6. To power on a node, enter:

```
rpower -n cls04 on
```

7. To query the power status of the nodes in a node group, enter:

```
rpower -N group1 query
```

Output is similar to:

```
cls01.ppd.pok.ibm.com on
cls02.ppd.pok.ibm.com on
cls05.ppd.pok.ibm.com on
```

8. To query the LCD status of a node with the **hmc** power method, enter:

```
rpower -l -n cls04 query
```

Output is similar to:

```
cls05.pok.ibm off LCD1 = 257
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

*/etc/opt/csm/system\_config/node ID*

The file containing the ID and encrypted password.

## Location

*/opt/csm/bin/rpower*

## See Also

- The **lsnode** and **systemid** commands.
- The *IBM CSM for Linux: Hardware Control Guide*.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## smsupdatenode Command

### Purpose

**smsupdatenode** - Updates open source RPM packages on CSM Linux nodes. The **smsupdatenode** command is called by the **updatenode** command.

### Synopsis

```
smsupdatenode [-h] [-v | -V] [-t | --test] [-q | --query [-c | --common]]
[--noinsdeps] [-r remote shell path] [-i | --install packagename,...] [-e |
--erase {--deps | --nodeps} packagename,...] [-p | --packages
packagename,...] [-a | -N nodegroups | [-n] node_list]
```

```
smsupdatenode [--path pkg_path] --copy {Attr=value ... | hostname}
```

### Description

The **smsupdatenode** command updates, installs or deletes open source RPM packages across a CSM cluster. There is no default target for the command; command line options must be provided to specify target machines. The **smsupdatenode** command runs only on Linux nodes. If any AIX nodes are specified on the command line, they are ignored.

**Note:** The **smsupdatenode** command cannot be used for Minimally Managed nodes – nodes on which CSM and RSCT have not been installed.

The **smsupdatenode** command updates RPM packages on the target nodes with RPM packages that are located in directories that correspond to the nodes' attributes located in `/csminstall/InstallOSName/InstallDistributionName/InstallDistributionVersion \ /InstallPkgArchitecture/updates`. RPM updates are installed only on nodes that have an older version of the same RPM already installed. However, if the RPM update has additional dependencies on RPM packages not installed, the **smsupdatenode** command will automatically install these dependencies if they are available in either the current **updates** directory or the corresponding `/csminstall/InstallOSName/InstallDistributionName/InstallDistributionVersion \ /InstallPkgArchitecture/RPMS` directory. This default dependency installation can be overridden using the **--noinsdeps** option, which instructs the **smsupdatenode** command not to install any dependencies.

The **smsupdatenode** command can also install new RPM packages on Linux nodes. The installation RPM packages should be placed in or linked to the `/csminstall/InstallOSName/InstallDistributionName/InstallDistributionVersion \ /InstallPkgArchitecture/updates` directory corresponding to the target node's configuration. Then run the **smsupdatenode -i** or **smsupdatenode --install** command with the RPM and target node names. The command installs the RPM packages on the target nodes, resolving dependencies from both the **/updates** and **/RPMS** directories.

The **-e** or **--erase** option instructs the **smsupdatenode** command to remove the specified RPM packages from the target nodes. There is no default dependency resolution for RPM deletion; you must either specify **--deps** to erase the RPM packages dependent on the target RPM packages, or **--nodeps** to erase only the target RPM.

The **smsupdatenode** command can also report the list of RPM packages installed on the target nodes. Since there are normally hundreds of RPM packages, the command breaks them down into two categories: **common** and **unique**. The **common** RPM packages are installed on all target nodes. The **unique** RPM packages include any RPM installed on a node that is not a part of the common set. The **smsupdatenode -q** or **--query** command returns the unique RPM packages installed on each target machine. Inclusion of the **-c** or **--common** option reports the common set of RPM packages.

The directory structure used by the **smsupdatenode** command is created during CSM node installation. While the directory structure requires that RPM packages be explicitly categorized by node attributes, symbolic links are allowed. So if you have an RPM that spans multiple node operating system configurations, you can make links to the RPM between the directory structures. All targets of symbolic links must reside under the **/csminstall/Linux** directory structure for NFS mounting purposes.

**Note:** If the *InstallPkgArchitecture* is **i386**, **i486**, **i586**, or **i686**, the architecture used in the *InstallPkgArchitecture* field of the directory structure above is **i386**.

If available, distribution RPM packages – all RPM packages provided by a Linux distribution CD – will be placed into the following directory during installation: **/csminstall/InstallOSName/InstallDistributionName/InstallDistributionVersion \ /InstallPkgArchitecture/RPMS**, for example, **/csminstall/Linux/RedHat/8.0/i386/RPMS/**. Updates or additions to these RPM packages should be placed in the corresponding directory: **/csminstall/InstallOSName/InstallDistributionName/InstallDistributionVersion \ /InstallPkgArchitecture/updates**, for example, **/csminstall/Linux/RedHat/8.0/i386/updates/**.

The **smsupdatenode** command uses **dsh** and the corresponding **DSH\_FANOUT** environment variable to control scalability.

In order to supply dependencies for RPM updates, you should place the RPM packages from the Operating System's CD in the corresponding **/csminstall/.../RPMS** directory. This is done by the **csmsetupks** and **csmsetupyast** commands if you do a full install on your nodes. However, if you are installing CSM-only nodes, you can use the **smsupdatenode** command to copy these RPM packages for you. This is done with the **--copy** (and optionally **--path**) option.

## Parameters

None.

## Options

**-a** Targets all nodes. Note that you cannot use the **-a** option with the **-n** option.

**-c** | **--common**

Specifies to also return the common set of RPM packages installed on all target nodes.

**--copy** {*Attr=value...* | *hostname*}

Copy the distribution CD-ROMs corresponding to the given attributes or *hostname* to the correct **/csminstall** directory. If you specify *Attr=value* pairs, they must be at the end of the command line. The valid attributes are:

## smsupdatenode

- *InstallDistributionName*
- *InstallDistributionVersion*
- *InstallPkgArchitecture*

If a *hostname* is specified, the distribution CD-ROMs, and destination directory are determined by the node's attributes.

### **-e** | **--erase** *packagename...*

Removes the given RPM packages. Either the **--deps** or **--nodeps** option must be provided before the list of packages to be removed. The **-i -p -e -q** options are mutually exclusive.

**--deps** Removes all packages dependent on the packages targeted for removal.

**--nodeps** Removes only the specified packages and leaves the dependent packages installed.

### **--file** *filename*

Specifies a file that contains a list of nodes names. If the specified file name is **-**, then the list is read from standard input. The file can contain multiple lines and each line can have one or more node names, separated by spaces.

**-h** Displays usage information.

### **-i** | **--install** *packagename...*

Installs the specified RPM packages. The absolute path is determined automatically by searching the directory structure. This option cannot be used with the **-p**, **-e**, or **-q** options.

### **-n** *node\_list*

Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.

Note that you cannot use the **-n** option with the **-a** option.

### **--noinsdeps**

Specifies to not install RPM dependencies.

### **-N** *nodegroups*

Specifies a node group or a comma-separated list of node groups.

### **-p** | **--packages** *packagename...*

Updates only the specified packages. The absolute path is determined automatically by searching the directory structure. This option cannot be used with the **-e**, **-i**, or **-q** options.

### **--path** *pkg\_path*

Specifies one or more directories, separated by colons, that contain copies of the distribution CD-ROMs. The default on a Linux Red Hat system is **/mnt/cdrom**. On a SuSE system, the default is **/cdrom**. This option may be used only with the **--copy** option.

### **-q** | **--query**

Queries all RPM packages installed on the target machines and returns the installed RPM packages that are unique to each target node. This option cannot be used with the **-e**, **-i**, or **-p** options.



**-r remote\_shell\_path**

Specifies the path to use for remote commands. This can either point to **rsh** or to a secure shell executable. If not specified with options, the remote shell defaults first to the value of the **DSH\_REMOTE\_CMD** environment variable, and then to the CSM *RemoteShell* attribute value (see the **csmconfig** command man page).

**-t | --test**

Reports what would be done by the command without making any changes to the target nodes.

**-v | -V**

Specifies verbose mode.

## Exit Status

- 0** The command has run successfully.
- 1** An error has occurred.
- 2** Could not NFS mount **/csminstall** on a node.
- 3** Another instance of the **smsupdatenode** command is currently running.
- 4** Could not query Managed node data because CTRMC is down.
- 10** **AutoUpdate** encountered some errors.
- 21** A host name or node group name resolution error occurred. Run the CTRMC probe and check the input to the **smsupdatenode** command.
- 22** One or more target nodes were unreachable. The Status is 0 or there is a problem with **dsh**.

## Security

The command requires root access to the cluster management server.

## Examples

1. To update all CSM Linux nodes with all RPM packages in the **/csminstall/InstallOSName/InstallDistributionName/InstallDistributionVersion/InstallPkgArchitecture/updates** directory, enter:  

```
smsupdatenode -a
```
2. To test what will be done when you update **webserver**, enter:  

```
smsupdatenode --test webserver
```
3. To install the latest version of FTP on **webserver**, make sure that **/csminstall/Linux/RedHat/8.0/i386/updates/ftp-0.17-7.i386.rpm** exists, then enter:  

```
smsupdatenode --install ftp-0.17-7.i386.rpm webserver
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Files

**/var/log/csm/smsupdatenode** Location of the **smsupdatenode** log file.

**smsupdatenode**

**Location**

`/opt/csm/bin/smsupdatenode`

**See Also**

The **updatenode** command.

**Author**

IBM CSM Development - cluster@us.ibm.com

---

## systemid Command

### Purpose

**systemid** - Stores the user ID and password required for internal programs to access remote hardware.

### Synopsis

```
systemid [-h] | [hostname | ip_address] | [[-s] -p power_method user_id] | [[-s]
host_name | ip_address user_id]
```

### Description

The **systemid** command stores the user ID and password for remote hardware. The hardware is identified by its IP address or host name. The password file name is the IP address resolved from the host name entered. If the host name is not resolvable, the password file is stored as the host name that was entered.

When running the **systemid** command, you will be prompted for the password to be stored with the specified user ID. If the command is run with no user ID it will return the IP address and user ID for the specified IP address, if one has been stored. If the command is run with no IP address, it will return all IP addresses and user IDs that have been previously saved. To change an existing saved user ID or password value, rerun the command with new values.

The **-p** option allows the creation of a default system ID entry that can be used for all hardware control points of a given power method. This can be useful if the login credentials for a group of hardware control points are the same. CSM software will first check for a unique system ID entry for the given node's hardware control point. If it does not find one, it will then check for a default system ID entry for the given node's power method.

### Parameters

<i>hostname</i>	Specifies the host name of the system where the ID resides.
<i>ip_address</i>	Specifies the IP address of the system where the ID resides.
<i>user_id</i>	Specifies the user name that will be defined. A user ID of <b>null</b> (case insensitive) will result in "no user ID." A password value of <b>null</b> (case insensitive) or just pressing the <b>Enter</b> key when prompted to enter a password, will result in "no password".

### Options

<b>-h</b>	Writes the command's usage statement to standard output.
<b>-p</b> <i>power_method</i>	Specifies the name of the <i>PowerMethod</i> that will be associated with the user ID and password that are entered. The <i>PowerMethod</i> value can be <b>xseries</b> , <b>hmc</b> , <b>apc</b> or <b>bmc</b> . The output <i>PowerMethod_#DEFAULT#</i> is returned.
<b>-s</b>	Sets the password on the specified node, hardware control point, or both when the <b>-p</b> option is used.

### Security

The command requires root access to the cluster management server.

## systemid

### Examples

1. To determine the hardware control points and user IDs specified, enter:  
systemid

The output would look similar to the following:

```
9.114.132.7      USERID
9.114.132.70    USERID
9.114.133.236   apc
9.114.67.60     hscroot
9.114.67.61     hscroot
```

**Note:** Each user ID is stored by the IP address of the hardware control point, if **systemid** was able to resolve the host name to an IP address. If it was not able to resolve the host name, then it stores the USERID with the text string of the name specified.

2. To store a system ID for a hardware control point with an IP address of 8.111.12.345 and a user ID of **hscroot**, enter:  
systemid 8.111.12.34 hscroot

You will be prompted for the password, as follows:

```
Password:
Verifying, please re-enter password:
systemid: Entry updated.
```

3. To store a system ID for all nodes with a *PowerMethod* of **hmc** and a user ID of **hscroot**, enter:  
systemid -p hmc hscroot

You will be prompted for the password, as follows:

```
Password:
Verifying, please re-enter password:
systemid: Entry updated.
```

### Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

### Location

**/opt/csm/bin/systemid**

### See Also

The **rconsole** and **rpower** commands.

### Author

IBM CSM Development - cluster@us.ibm.com

---

## uninstallms Command

### Purpose

**uninstallms** - Uninstalls the Cluster Systems Management (CSM) management server software.

### Synopsis

**uninstallms** [-h] [-u] [-f] [-v | -V]

### Description

This **uninstallms** command uninstalls the CSM management server software. The command performs the following tasks:

- Removes node group definitions.
- Removes cluster node definitions using the **rmnode** command.
- If the **-u** option is specified, the command performs clean up on the node by running the **rmnode -u** command. This clean up includes removing CSM log files and uninstalling CSM packages from the nodes. See the **rmnode** command for details.
- Removes predefined conditions.
- Uninstalls CSM packages that were installed on the management server.
- Removes CSM log files.

The **uninstallms** command will not uninstall any CSM packages that were automatically installed with the operating system, or any open source prerequisites that may have been installed. The command will not remove any of the **/csminstall** directories since they may contain data not pertaining to CSM. If the management server is also a node (has **csm.client** installed), **uninstallms** will not uninstall **csm.client**, **rsct.core**, **rsct.core.utils** or **src**. If **rsct.basic** is installed (a prerequisite for GPFS and other products), **uninstallms** will not uninstall **rsct.basic**, **rsct.core**, **rsct.core.utils** or **src**.

If the **addlcsext** command has been run on the CSM management server, **rmlcsext -A** must be run before running the **uninstallms** command.

### Parameters

None.

### Options

- f** Forces an uninstall of the management server without prompting.
- h** Displays usage information.
- u** Runs **rmnode -u** for each node before uninstalling the management server.
- v | -V** Specifies to return verbose messages.

### Exit Status

- 0** The command has run successfully.
- 1** An error has occurred.

**uninstallms**

## Security

The command requires root access to the cluster management server.

## Examples

To uninstall the CSM management server software, enter:

```
uninstallms -u
```

## Implementation Specifics

This command is part of IBM Cluster Systems Management (CSM) software.

## Location

*/opt/csm/bin/uninstallms*

## See Also

The **addlcsext**, **rmlcsext**, and **rmnode** commands.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## updateisvr Command

### Purpose

**updateisvr** - Updates CSM installation files across multiple install servers.

### Synopsis

```
updateisvr [-h] [-v | -V] {-a | -N nodegroups | --file filename | [-nnode_list]}
```

### Description

The **updateisvr** command updates CSM installation files across multiple install servers. Use the **updateisvr** command after you have set the *InstallServer* attribute for the desired nodes. It distributes all the needed files from the **/csminstall** directory to the install servers. It is important to do the first sync of the servers using the **updateisvr** command, because the first sync could require a considerable amount of time. After the first sync, the install servers are maintained by the CSM commands that utilize them.

**Note:** This command is not supported for Linux on pSeries clusters.

You must set up remote shell access to all install servers (you must be able to run a **ssh** command to all the servers) in order for the **updateisvr** command to work.

**Note:** Install servers are currently only supported for Red Hat and Red Hat AS clusters.

### Parameters

None.

### Options

**-a** Updates all install servers used by nodes in this cluster. Note that you cannot use the **-a** option with the **-n** option.

**-N** *nodegroups*  
Specifies a comma-separated list of node groups on which to run the command.

**--file** *filename*  
Updates all install servers listed in the file.

**[-n]** *node\_list*  
Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. Host names must be specified as positional arguments or preceded by the **-n** flag. For information about specifying node ranges, see the **noderange** man page.

Note that you cannot use the **-n** option with the **-a** option.

### Security

The command requires root access to the cluster management server.

### Examples

1. To update all of the install servers that are used by nodes in the cluster, enter:

## updateisvr

```
updateisvr -a
```

2. To update all of the install servers in node groups nodegroup3, nodegroup4 and nodegroup5, enter:

```
updateisvr -N nodegroup3,nodegroup4,nodegroup5
```

## Files

<code>/opt/csm/bin/dsh</code>	Location of the <b>dsh</b> command.
<code>/var/log/csm/updateisvr.log</code>	Location of the <b>updateisvr</b> log file.

## Location

```
/opt/csm/bin/updateisvr
```

## See Also

The *IBM CSM for Linux: Planning and Installation Guide*.

## Author

IBM CSM Development - cluster@us.ibm.com



## updatenode Command

### Purpose

**updatenode** - Updates or installs CSM packages, Linux RPM packages, and CFM configuration files across the cluster.

### Synopsis

```
updatenode [-h] [-v | -V] [-f] [-t | --test] [-k] [--nopwd] [-F | --cfm]
[--cfmoptions "cfm_options"] [-S | --sms] [--smsoptions "sms_options"] {-a |
-P | -M | -N nodegroups | --file filename | [-n] node_list}
```

### Description

The **updatenode** command runs on the management server to add a node to the cluster or update existing CSM nodes. The command can be used to add a node to the cluster, run user-provided customization scripts, and distribute configuration files (CFM). For Linux nodes it may also be used to install CSM software and to apply Linux software maintenance (SMS).

By default, the **updatenode** command adds a node to the cluster, runs user-provided customization scripts (if provided), and distributes configuration files using CFM. For Linux nodes it also installs or updates CSM software and performs Linux software maintenance using SMS. The **updatenode** command can be instructed to perform only some of these tasks using command line options.

- If run with either the **-S**, **--sms** or **--smsoptions** options, the **updatenode** command only performs software maintenance on Linux target nodes.
- If run with either the **-F**, **--cfm** or **--cfmoptions** options, the **updatenode** command only distribute configuration files.

The **updatenode** command can be used to install CSM on a PreManaged node. During this process the PreManaged node is added to the CSM cluster and converted to a Managed node. For convenience, the **updatenode** command can also be run using the **definnode -U** command, which invokes the **updatenode -P** command after the **definnode** command has completed.

**Note:** When updating or installing MinManaged nodes (nodes on which CSM and RSCT have not been installed and have minimal CSM management), the **updatenode** command only sets up the remote environment. It does not install any packages.

If the **addlcsext** command has been run on the CSM management server, **updatenode** ensures that **addlcsext** is automatically run on the specified Managed and PreManaged nodes.

### Parameters

None.

### Options

**-a** Performs updates on all nodes. You cannot use the **-a** option with the **-M**, **-P**, **-N**, or **-n** options.

**--cfmoptions** "*cfm\_options*"

Forwards options to the **cfmupdatenode** command. This option implies the

## updatenode

[**-F** | **--cfm**] option. All options following **--cfmoptions** must be enclosed within double quotes. The following options are allowed: [**-b**], [**-y**] | [**-c**], [**-r remote\_shell\_path**], [**-t timeout**], [**-M max\_child\_process**], [**-d distfile\_location**], [**-F filename**].

- f** Forces the updates of packages and performs a binary check on configuration files.
- file filename**  
Specifies a file that contains a list of node names. If the file name specified is **-**, then the list is read from standard input. The file can contain multiple lines and each line can have one or more node names, separated by spaces.
- F** | **--cfm**  
Updates the configuration files from the **/cfmroot** directory. The **-I**, **-S**, and **-F** options are the **updatenode** defaults. When individual command line options are used, the **updatenode** command performs only the actions specified by the given options.
- h** Displays usage information.
- k** Exchanges RSCT public keys, host name, and node identifier between the node and the management server. Use this option only if the host name, node identifier, or public key of either machines has changed. At the time this option is used, you must trust your network to ensure that a rogue machine is not going to insert its public key in place of the public key of the real node.
- M** Performs updates on all nodes with *Mode* attribute values set to **Managed**. You cannot use the **-M** option with the **-a**, **-P**, **-N**, or **-n** options.
- nopwd**  
Specifies to not prompt for the node password even if **rsh** or **OpenSSH** is not configured. If you do not use **--nopwd** and the remote shell is not configured, the **updatenode** command prompts for the node root password and attempts configuration.
- n node\_list**  
Specifies a comma or space-separated list of node host names, IP addresses, or node ranges on which to run the command. Space-separated lists must be inside double quotes. For information about specifying node ranges, see the **noderange** man page.  
  
Note that you cannot use the **-n** option with the **-a**, **-M**, or **-P** options.
- N nodegroups**  
Specifies a comma-separated list of node groups on which to run the command. This option cannot be used with the **-a**, **-P**, or **-M** options.
- P** Performs updates on all nodes with a *Mode* attribute value of **PreManaged**. You cannot use the **-P** option with the **-a**, **-M**, **-N**, or **-n** options.
- S** | **--sms**  
Installs or updates the Linux RPM packages stored under the **/csminstall** directory corresponding to the target node. The **-I**, **-S**, and **-F** options are the **updatenode** defaults. When individual command line options are used, the **updatenode** command performs only the actions specified by the given options.
- smsoptions "sms\_options"**  
Forwards options to the **smsupdatenode** command. This option runs the

`[-S | --sms]` option. All options following `--smsoptions` must be enclosed within double quotes. The option values can include `[-i | --install packagenames]`, `[-e | --erase [--deps] | [--nodeps]] packagenames]`, or `[-p | --packages] packagenames]`, where *packagenames* is a comma-separated list.

`-t | --test`

Only reports what would be done by the command. Does not perform any updates.

`-v | -V`

Specifies verbose mode.

## Environment Variables

### CSM\_FANOUT

Sets the fanout value for the **updatenode** command operations. The default value is 32.

## Diagnostics

- 0 The command completed successfully.
- 1 A usage error occurred.
- 2 The **makenode** script did not complete successfully.
- 3 CFM did not complete successfully.
- 5 Both the **makenode** script and CFM did not complete successfully.
- 6 SMS did not complete successfully.
- 8 Both the **makenode** script and SMS did not complete successfully.
- 9 Both CFM and SMS did not complete successfully.

## Security

The command requires root access to the cluster management server.

## Examples

1. To install all PreManaged nodes, enter:  

```
updatenode -P
```
2. To update all Managed nodes automatically every morning at 0800, add the following line to your **crontab** file:  

```
0 8 * * * /opt/csm/bin/updatenode -M
```
3. To test the **updatenode** command on the **FTP\_server** node group, enter:  

```
updatenode -t -N FTP_server
```

## Files

- /cfmroot** The directory on the management server that contains the cluster's configuration files.
- /opt/csm/bin/cfmupdatenode** Location of the **cfmupdatenode** command.
- /opt/csm/bin/smsupdatenode** Location of the **smsupdatenode** command.

## updatenode

**/var/log/csm/updatenode.log** Location of the log file on the management server for the **updatenode** command. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4.

**/var/log/csm/install.log** Location of the log file on each node containing installation information. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4.

## Location

**/opt/csm/bin/updatenode**

## See Also

The **addlcsext**, **cfmupdatenode**, **definenode**, **dsh**, and **smsupdatenode** commands.

The *IBM CSM for Linux: Administration Guide*.

## Author

IBM CSM Development - cluster@us.ibm.com

---

## Appendix A. CSM log files

See the *IBM CSM for Linux: Hardware Control Guide* for the CSM hardware control log files.

Log file	Description
<code>/var/log/csm/cfmerror.log</code>	Location of CFM error logs.
<code>/var/log/csm/cfmchange.log</code>	Location of the log, on the management server, that contains the dates when files were updated.
<code>/var/log/csm/installms.log</code>	Location of the log file on the management server for the <b>installms</b> command. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4.
<code>/var/log/csm/install.log</code>	Location, on each node, of the log output from the <b>updatenode</b> command. This log file contains installation information. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4.
<code>/var/log/csm/updatenode.log</code>	Location of the log file on the management server for the <b>updatenode</b> command. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4.
<code>/var/log/csm/csmsetupks.log</code>	Location of the log file on the management server for the <b>csmsetupks</b> command. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4.
<code>/var/log/csm/installnode.log</code>	Location of the log file on the management server for the <b>installnode</b> command. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4.
<code>/var/log/csm/csmsetupyast.log</code>	Location of the log file on the management server for the <b>csmsetupyast</b> command. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4.



---

## Appendix B. Glossary

**action.** The part of the event response resource that contains a command and other information about the command.

**APAR.** Authorized Program Analysis Report. A report of a problem caused by a suspected defect in a current unaltered release of a program.

**ASM.** See ISMP.

**attribute.** Attributes are either persistent or dynamic. A resource class is defined by a set of persistent and dynamic attributes. A resource is also defined by a set of persistent and dynamic attributes. Persistent attributes define the configuration of the resource class and resource. Dynamic attributes define a state or a performance-related aspect of the resource class and resource. In the same resource class or resource, a given attribute name can be specified as either persistent or dynamic, but not both.

**Audit Log.** A log file containing a record of system events and responses.

**authentication.** The process of validating the identity of an entity, generally based on user name and password. However, it does not address the access rights of that entity. Thus, it simply makes sure that a user is who he or she claims to be.

**authorization.** The process of granting or denying access to an entity to system objects or resources, based on the entity's identity.

| **AutoYaST.** The part of the SuSE and SuSE SLES  
| operating systems related to Linux OS installation. See  
| also the **csmssetupyast** command.

**BIOS.** Basic Input/Output System. Microcode that controls basic hardware operations such as interactions with diskette drives, fixed-disk drives, and the keyboard.

**BladeCenter.** IBM consolidated high performance eServer hardware racks, which can be used in a CSM cluster.

**BladeCenter chassis.** A chassis that can hold up to 14 hot-swappable blade servers.

**Blade server.** An independent server containing one or more processors and associated memory, disk storage and network controllers, and running its own operating system and software.

| **BMC.** The baseboard management controller (**bmc**) is  
| firmware on the eServer 325 NIC card that handles all  
| network traffic. If it detects a hardware control  
| command, the **bmc** accepts and processes the  
| command; otherwise, it ignores it and forwards it to the  
| node.

| **bmc.** **bmc** is the *PowerMethod* attribute value for  
| nodes on eServer 325 servers.

**CFM.** The Configuration File Manager maintains files that are common across all nodes in a cluster.

**client.** Client applications are the ordinary user interface programs that are invoked by users or routines provided by trusted services for other components to use. The client has no network identity of its own: it assumes the identity of the invoking user or of the process where it is called, who must have previously obtained network credentials.

**client node.** In CSM, all nodes except the management server are considered client nodes. In a client/server model, the client system sends requests to a server system, who fulfills the request and returns status.

**cluster.** A group of servers and other resources that act like a single system and enable high availability and, in some cases, load balancing and parallel processing.

**cluster hardware control point.** See hardware control point.

**clustering.** The use of multiple computers (such as UNIX workstations), multiple storage devices, and redundant interconnections to form what appears to users as a single highly-available system. Clustering can be used for load balancing, for high availability, and as a relatively low-cost form of parallel processing for scientific and other applications that lend themselves to parallel operations.

**Cluster Systems Management.** IBM Cluster Systems Management software for AIX and Linux is designed for simple, low-cost management of distributed and clustered IBM eServers in technical and commercial computing environments. CSM, included with the Cluster 1350 (Linux) and optional with the IBM Cluster 1600 (AIX), simplifies cluster administration by providing management from a single point-of-control.

**cluster VLAN.** The cluster Virtual LAN (VLAN) connects nodes to each other and to the management server through an Ethernet connection. Installation and CSM administration tasks such as running **dsh** are done on the cluster VLAN.

**coexistence.** The ability of two different pieces of software, running either on the same machine or on machines that are interconnected, to function together. For example, an AIX node and a Linux node coexist in a mixed cluster having an AIX management server.

**condition.** A certain state of a node resource that can be monitored.

**console.** The main operating system display station. Synonym for system console.

**console server.** The hardware device through which the management server opens a remote console session for a node.

**consumability.** Uses the **snmptrap** command to generate traps containing ERRM event information that can be sent to an SNMP manager.

**CSM.** See Cluster Systems Management.

**CSM database.** A repository of cluster, node, and node group information that is created and used by CSM.

**CSM GUIs.** Graphical User Interfaces (GUIs) available for running CSM functions: IBM Web-based System Manager, SMIT, and DCEM GUIs.

**CSM-only installation.** The process of installing only CSM on the nodes, as opposed to a full installation, which involves installing both CSM and the operating system on the nodes.

**CSM plug-ins.** IBM Web-based System Manager GUI plug-ins, which provide an interface for monitoring and managing one or more CSM clusters.

**DCEM.** Distributed Command Execution Manager is a GUI that can run commands on multiple cluster nodes simultaneously.

**distribution.** One of the Linux operating systems used with CSM. For example, Red Hat, SuSE, or SuSE SLES.

**domain.** (1) A set of network resources (such as applications and printers, for example) for a group of users. A user logs in to the domain to gain access to the resources, which could be located on a number of different servers in the network. (2) A group of server and client machines that exist in the same security structure. (3) A group of computers and devices on a network that are administered as a unit with common rules and procedures. Within the Internet, a domain is defined by its Internet Protocol (IP) address. All devices that share a common part of the IP address are in the same domain.

**Domain Management Server resource manager (IBM.DMSRM).** Controls the Managed node (IBM.ManagedNode) resource class and the node group (IBM.NodeGroup) resource class.

**device driver.** (also, **driver**, **kernel module**.) A software program that interacts with a particular hardware device or with other software. Downloading or building kernel modules may be required to install an operating system on certain hardware. A different kernel module is required for each version of the Linux kernel.

**dsh.** A distributed shell program - a mechanism to issue commands to all systems in a network, in parallel.

**dynamic attribute.** A node attribute with a value that can change over time, such as node power status.

**dynamic node group.** A variable node group consisting of nodes with specific attribute values.

**ERRM.** RSCT Event Response Resource Manager controls events and responses on cluster nodes.

**ESP.** The Equinox Ethernet Serial Provider allows you to place COM serial ports anywhere on a local or remote LAN segment. The ESP units communicate with an Equinox SuperSerial NT driver located on Windows NT Server and Workstation systems. Ports on the ESP appear to the servers as standard COM ports as if they were right on the servers' system bus. All the facilities and functions of Windows NT and of application programs are fully available to these LAN-resident COM ports. The ESP units and the LAN are "transparent".

**Ethernet.** (1) Ethernet is the standard hardware for TCP/IP local area networks in the UNIX marketplace. It is a 10-megabit per second baseband type LAN that allows multiple stations to access the transmission medium at will without prior coordination. The Ethernet avoids contention by using carrier sense and deference, and resolves contention by collision detection (CSMA/CD). (2) A passive coaxial cable whose interconnections contain devices or components, or both, that are all active. It uses CSMA/CD technology to provide a best-effort delivery system.

**event.** Occurs when the event expression of a condition evaluates to True. An evaluation occurs each time an instance of a dynamic attribute is observed.

**event expression.** A definition of the specific state when an event is true.

**event response.** One or more actions as defined by the event response resource manager (ERRM), that take place in response to an event or a rearm event.

**fanout.** The number of systems or processors that are to receive software updates or communications simultaneously. For CSM, this is controlled by the environment variable **CSM\_FANOUT**. The **DSH\_FANOUT** environment variable is used by the **dsh** command to control the number of nodes on which to simultaneously run a remote command.

**fileset.** For AIX, a collection of files, usually used to install a piece of software. The equivalent Linux term is package.

**fix.** A correction or enhancement to software.

**full installation.** The process of installing both the CSM software and the operating system on the nodes



of the cluster, as opposed to installing only CSM on the nodes, or installing only the operating system on the nodes.

**GPFS.** The IBM General Parallel File System (GPFS) for AIX and Linux allows users shared access to files that may span multiple disk drives on multiple nodes. It offers many of the standard AIX file system interfaces, allowing most applications to run without modification or recompiling. AIX file system utilities are also supported by GPFS.

**hardware control point.** The hardware device through which the management server controls node hardware.

**Hardware Control resource manager.** The IBM Hardware Control resource manager manages the IBM.NodeHwCtrl and IBM.HwCtrlPoint resource classes.

**Hardware Management Console.** The IBM Hardware Management Console for pSeries is an installation and service support processor that runs only the HMC software.

**HMC.** See Hardware Management Console.

| **hmc.** **hmc** is the *PowerMethod* and *ConsoleMethod*  
| attribute value for nodes on HMC-attached pSeries  
| servers

**hostmap file.** See hostname mapping file.

**host name.** (1) A name assigned to a computer connected to a network. The use of this term can be ambiguous as it can refer to either the short form name of the computer (see short host name), or the fully qualified name of the computer (see long host name). (2) The Internet address of a machine in the network. Also known as host ID.

**hostname mapping file.** A file containing a list of host names and associated hardware control information. This file can be created by the **lshwinfo** command and used as input to the **definnode** command.

**IBM.DMSRM.** See Domain Management Server resource manager.

**IBM.HWCTRLRM.** See Hardware Control resource manager.

**IBM.NodeHwCtrl.** See Node Hardware Control Resource Class.

**ISMP.** Integrated System Management Processor - a computer within a computer, the ISMP performs systems management tasks that help manage and maintain the health of your server. Integrated into select xSeries servers, the ISMP continuously monitors your system and notifies you of potential failures.

Through IBM Director, the ISMP alerts you to changes in system temperature, voltage, fan redundancy, memory and hard-drive performance. It also provides

configuration management benefits with features like remote firmware updates, remote power control, and Automatic Server Restart (ASR).

On Linux, the IBM Integrated System Management Processor (ISMP) device monitors and provides remote power control for xSeries servers. ISMPs are also referred to as ASMs.

**kernel.** The essential component of the Linux and AIX operating system. The kernel is responsible for critical OS functions such as resource allocation, low-level hardware interfaces, and security. Installation on certain hardware, certain Linux distributions, or certain device drivers might require a minimum kernel version.

**Kickstart.** On Linux, part of the Red Hat operating system used to help install Red Hat. Using Kickstart, a system administrator can create a single file containing the answers to all the questions that would normally be asked during a typical Red Hat Linux installation.

Kickstart files can be kept on single server system, and read by individual computers during the installation. This installation method can support the use of a single Kickstart file to install Red Hat Linux on multiple machines, making it ideal for network and system administrators. For CSM, see the **csmssetupks** command.

**kscfg.tmpl.** This file is the template used by the **csmssetupks** command to create a Kickstart configuration file for each Linux node. The template is located in **/opt/csm/install/kscfg.tmpl**.  
*InstallDistributionNameInstallDistributionVersion*.

The Kickstart configuration file generated by **csmssetupks** from the template contains configuration information gathered by Kickstart during installation of the Linux operating system. The **kscfg.tmpl** file can be used as is, or modified. See the sample template in the Appendix of *IBM CSM for Linux: Software Planning and Installation Guide* for instructions on how to properly modify the template.

**ksh.** Korn shell

**license key file.** A file containing keys (passwords) necessary to run CSM.

**license use key.** A key (password) that is required to run CSM. A license key file, containing license use keys, is included with the CSM package.

**Linux node.** One instance of a Linux operating system running on IBM xSeries hardware.

**long host name.** A fully-qualified host name (for example, node15.pok.ibm.com).

**LPAR.** Logical partition. The partitioning of an operating system and its associated resources, such as memory, to give the appearance and functionality of more than one operating system.

**MAC address.** The Media Access Control address is a hardware address that uniquely identifies each node of a network. On a local area network (LAN) or other network, the MAC address is the computer's unique hardware number. On an Ethernet LAN, it is the same as the computer's Ethernet address. CSM only uses the MAC address of the network adapter used to do network boot and installation – the network adapter on the cluster VLAN.

**machine architecture.** The type of hardware for a specific Linux node on xSeries hardware. For CSM, the machine architecture is specified by the *InstallPkgArchitecture* node attribute, and must be provided for hardware control. Currently, CSM supports i386 machine architecture only. However, i486, i586, and i686 processors can be used provided they are defined with an *InstallPkgArchitecture* of i386.

**Managed node.** A node in a CSM cluster under the control of the management server. This node has a Mode attribute of "Managed". The **updatenode** command converts PreManaged nodes to Managed nodes.

**management control point.** See management server.

**management domain.** A set of nodes that are configured for management by the Clusters Systems Management (CSM) licensed program. Such a domain has a management server that is used to administer a number of Managed nodes. Only management servers have knowledge of the whole domain. Managed nodes only know about the servers managing them; they know nothing of each other.

**management server.** A node with CSM cluster management server software installed.

**management VLAN.** The management Virtual LAN (VLAN) connects the management server to the cluster hardware through an Ethernet connection. For optimal security, the management VLAN must be restricted to hardware control points, remote console servers, the management server, and root users. Routing between the management VLAN and cluster or public VLANs could compromise security on the management VLAN.

**migration.** The process of moving to a later software version.

**mixed cluster.** A CSM cluster with an AIX 5L management server and both AIX and Linux nodes.

**NFS.** A distributed file system that allows users to access files and directories located on remote computers and treat those files and directories as if they were local. NFS allows different systems (UNIX or non-UNIX), different architectures, or vendors connected to the same network, to access remote files in a LAN environment as though they were local files.

**node.** One operating system image. See **Managed node**.

**node attribute.** Pieces of information that make up a node definition. For a CSM node, these attributes must be defined in the CSM database. See the **nodeattributes** man page for more details.

**node attribute template.** A worksheet used by the system administrator to record the attributes assigned to the nodes.

**node configuration template.** A worksheet used by the system administrator to record details of the node configuration.

**nodedef file.** See node definition file.

**node definition file.** A file containing a stanza of information for defining each node in a cluster. The information about each node is of the form *Attr=value*, such as *InstallOSName=AIX*. This file can be used by the **definnode** command. See the man page for **nodedef** for more details.

**node group.** Nodes having similar attribute values and defined as a group to facilitate node management.

**Node Hardware Control Resource Class (IBM.NodeHwCtrl).** Provides support for powering a node on and off, resetting a node, querying the power status of a node, resetting the node's service processor, and resetting the node's hardware control point.

**null value.** Empty, having no value, containing nothing.

**Open source software.** Any program whose source code is made available for use or modification as users or other developers see fit. Open source software is usually developed as a public collaboration and made freely available.

**OpenSSH.** For Linux, OpenSSH is a free version of the SSH protocol suite of network connectivity tools. For AIX, OpenSSH encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other network-level attacks. Additionally, OpenSSH provides a myriad of secure tunneling capabilities, as well as a variety of authentication methods. See the OpenSSH Web site at <http://www.openssh.com>.

**package.** For Linux, a collection of files, usually used to install a piece of software. The equivalent AIX term is **fileset**. For Linux, a package is also referred to as an RPM (Red Hat Program Manager) package.

**partition.** (1) A logical division of storage on a fixed disk. (2) A fixed-size division of storage. (3) a group of non-overlapping nodes that act as a logical system.

**persistent attribute.** A node attribute with a value that does not change without manual user input, such as node name.

**port number.** A port number is a way to identify a specific process to which an Internet or other network message is to be forwarded when it arrives at a server. For the TCP/IP and UDP protocols, a port number is a 16-bit integer that is put in the header appended to a message unit. This port number is passed logically between client and server transport layers, and physically between the transport layer and the Internet Protocol layer.

**predefined condition.** A condition whose definition is supplied by the RMC portion of RSCT. Predefined conditions are used to monitor certain system events, and may be customized for a particular installation.

**predefined dynamic node group.** A node group whose members all have a certain attribute set to a certain value. The definitions of these groups are shipped with CSM. For example, the **LinuxNodes** group consists of those nodes whose *InstallOSName=Linux*.

**predefined response.** A response whose definition is supplied by the RMC component of RSCT. Predefined responses are defined to take action when a certain condition becomes true. They may be customized for a particular installation.

**PreManaged node.** A node that is part of a CSM cluster, but has not yet been put under the control of the management server. Such a node has a Mode attribute of "Premanaged". The **updatenode** command converts PreManaged nodes to Managed nodes.

**probe.** Diagnostic software that assesses the functionality of a single machine at a time.

| **pSeries.** IBM eServer hardware that runs the AIX 5L  
| and Linux operating systems.

**public VLAN.** The public Virtual LAN (VLAN) connects the cluster nodes and management server to the site network. Applications are accessed and run on cluster nodes over the public VLAN. The public VLAN can be connected to nodes through a second Ethernet adapter in each node, or by routing to each node through the Ethernet switch.

**rconsole.** The remote console command. See remote console.

**rearm event.** An event that occurs when the rearm expression for a condition evaluates to True.

**rearm expression.** An expression that generates an event which alternates with an original event in the following way: the event expression is used until it is true; then, the rearm expression is used until it is true; then, the event expression is used. The rearm expression is commonly defined as the inverse of the

event expression. It can also be used with the event expression to define an upper and lower boundary for a condition of interest.

**Red Hat.** Red Hat is a software company in the business of assembling open source components for the Linux operating system and related programs into a distribution package.

**Red Hat Linux.** A version of Linux produced by Red Hat Inc.

**remote command.** A command issued on the management server that is intended to run on one of the cluster nodes.

**remote console.** From the management server, access to the operator console of one or more CSM nodes. See the **rconsole** command.

**remote hardware control.** Management server control of cluster node hardware.

**remote power.** Management server control of the following CSM node hardware characteristics: power on or off, query power status, reboot, and reset of the service processor. See the **rpower** command.

**remote shell.** When using the **dsh** command, the shell where the remote command will run. Also, the shell set up on each node during installation. In CSM, the *RemoteShell* attribute value specifies which remote shell is used. The default value on AIX is **/usr/bin/rsh**. The default value on Linux is **/usr/bin/ssh**.

**resource.** An entity in the system that provides a set of services. Examples of hardware entities are processors, disk drives, memory, and adapters. Examples of software entities are database applications, processes, and file systems. Each resource in the system has one or more attributes that define the state of the resource.

**resource class.** A group of resources that have attributes, actions, and other characteristics of the resource class in common.

**resource manager.** A standalone daemon that maps resource and resource class abstractions into calls and commands for one or more specific types of resources.

**response.** An automated response to a node resource condition.

**RMC.** The IBM Resource Monitoring and Control component of RSCT, which monitors and controls cluster nodes.

**RPM packages.** Software and updates for Linux nodes.

**rpower.** The remote power command. See remote power.

**RSA.** The IBM Remote Supervisor Adapter (RSA) is the hardware control point for xSeries servers.

**RSCT.** IBM Reliable Scalable Cluster Technology is a set of software components that together provide a comprehensive clustering environment for AIX and Linux. RSCT is the infrastructure used by a variety of IBM products, including CSM, to provide clusters with improved system availability, scalability, and ease of use.

**rsh.** A variant of the **rlogin** command that invokes a command interpreter on a remote UNIX machine and passes the command line arguments to the command interpreter, skipping the LOGIN step completely.

**Server File Repository.** A directory on the management server named **/cfmroot**, which contains the cluster configuration files

**servers.** Hardware that has server programs running in the background on the OS without a user's inherited credentials. A server must acquire its own network identity to get to access other trusted services.

**service processor.** A computer attached to a processor, whose sole function is to control the hardware and provide diagnostic support.

**shell.** The shell is the primary user interface for the UNIX operating system. It serves as command language interpreter, programming language, and allows foreground and background processing. Implementations of the shell concept include Bourne, C, and Korn.

**short host name.** A host name that contains only the local identifier.

**SIS.** System Installation Suite - On Linux, an open source product that helps you install and configure SuSE and SuSE SLES. For use with CSM, see the **csmssetupsis** command.

**SMS.** Software Maintenance System maintains RPM packages on Linux nodes from an AIX or Linux management server.

**SNMP.** Simple Network Management Protocol. (1) An IP network management protocol that is used to monitor attached networks and routers. (2) A TCP/IP-based protocol for exchanging network management information and outlining the structure for communications among network devices.

**ssh.** Secure Shell, sometimes known as Secure Socket Shell, is a Unix-based command interface and protocol for securely getting access to a remote computer.

**static node group.** A node group consisting of nodes specified by the user.

**SuSE.** SuSE is a privately owned German company whose mission is to promote open source development and General Public License distribution and to be a Linux distribution provider. SuSE assembles open source components for the Linux operating system and related programs into a selection of distribution packages.

**SuSE Linux.** A version of Linux produced by SuSE, Inc.

**SuSE SLES.** SuSE Linux Enterprise Server is a server operating system for professional deployment in heterogeneous IT environment of all sizes and sectors.

**update.** Software fixes periodically installed on a system.

**visual monitoring.** An icon-based method for monitoring a CSM cluster.

**VLAN.** Virtual LAN - Virtual Local Area Network. A division of a local area network by software rather than by physical arrangement of cables. Division of the LAN into subgroups can simplify and speed up communications within a workgroup. Switching a user from one virtual LAN to another via software is also easier than rewiring the hardware.

**xCAT.** xCAT (Extreme Cluster Administration Toolkit) is a tool kit that can be used for the deployment and administration of Linux clusters. Its features are based on user requirements, and many of its features take advantage of IBM xSeries hardware.

**xSeries.** IBM eServer hardware based on the Intel architecture.

| **yastcfg XML file.** Template used by the  
| **csmssetupyast** command to create an AutoYaST  
| configuration file for each Linux node. The template is  
| located in  
| **opt/csm/install/yastcfg.InstallDistributionName**  
| **InstallDistributionVersion-Arch.xml**.

---

## Appendix C. Bibliography

This Bibliography helps you find documentation related to Cluster Systems Management (CSM).

---

### Related information

The following references contain information about IBM Cluster Systems Management for Linux:

- *IBM CSM for Linux: Software Planning and Installation Guide*, SA22–7853
- *IBM CSM for Linux: Administration Guide*, SA22–7873
- *IBM CSM for Linux: Hardware Control Guide*, SA22–7856
- *IBM CSM for Linux: Command and Technical Reference*, SA22–7933

The following references contain information about Reliable Scalable Computing Technology (RSCT) for Linux:

- *IBM RSCT for Linux: Administration Guide*, SA22–7892
- *IBM RSCT for Linux: Technical Reference*, SA22–7893
- *IBM RSCT for Linux: Messages*, GA22–7894
- *IBM RSCT for Linux: Group Services Programming Guide and Reference*, SA22–7888

---

### Obtaining publications

The CSM and RSCT for Linux publications are available at either of the following Web sites:

- <http://www.ibm.com/servers/eserver/clusters/library>
- <http://www.ibm.com/shop/publications/order>

The @server Cluster 1350 InfoCenter is available at:  
<http://publib.boulder.ibm.com/cluster/>.

### Redbooks

The IBM International Technical Support Organization (ITSO) publishes Redbooks related to CSM.

- Linux Clustering with CSM and GPFS

For a current list, see the IBM Redbooks Web site at: <http://www.ibm.com/redbooks>.

### Other CSM information

See the following references for information related to CSM:

Information about CSM	Location
Service information (fixes and updates)	<a href="http://techsupport.services.ibm.com/server/cluster">http://techsupport.services.ibm.com/server/cluster</a>
CSM driver downloads	<a href="http://techsupport.services.ibm.com/server/cluster2/fixes/csmdriverdownload.html">http://techsupport.services.ibm.com/server/cluster2/fixes/csmdriverdownload.html</a>
README file	<b>/opt/csm/README/csm.README</b>
Documentation Errata	<a href="http://publib.boulder.ibm.com/clresctr/docs/csm/docerrata.html">http://publib.boulder.ibm.com/clresctr/docs/csm/docerrata.html</a>



Information about CSM	Location
Frequently Asked Questions (FAQ)	<a href="http://techsupport.services.ibm.com/server/cluster/tips/csm_faq.html">http://techsupport.services.ibm.com/server/cluster/tips/csm_faq.html</a>
Read This First document	<a href="http://www.ibm.com/servers/eserver/clusters/library">http://www.ibm.com/servers/eserver/clusters/library</a>

---

## Getting XCAT tools

If you are an XCAT user, you may find the IBM alphaWorks® *Enhanced Cluster Tools (ECT) for Linux* Web site useful. It is a repository of tools that complement CSM and enhance the management of Linux clusters. The ECT for Linux site provides tools that supplement CSM features such as remote access to hardware inventory and vitals, remote access to server processor logs, and support for the ELS console server. The ECT site includes early versions of tools that will eventually be merged into the CSM product, prototypes of new technology that is being investigated for CSM, and tools for specific vertical market segments. The ECT for Linux site is located at <http://www.alphaworks.ibm.com/tech/ect4linux>.

---

## Getting help from IBM

CSM mailing list information is available at <http://www.ibm.com/developerworks/oss/mailman/listinfo/csm>. E-mail sent to this mailing list at [csm@www-124.ibm.com](mailto:csm@www-124.ibm.com) is monitored by the CSM development team, providing a mechanism for users to ask questions and resolve problems. If the mailing list does not solve your problem, then you can send a note directly to the CSM development team at [cluster@us.ibm.com](mailto:cluster@us.ibm.com), or call IBM Support at 1-800-IBM-SERV.

Before you call for help, check to see if all the latest service has been applied to your system. Then, see the diagnosis section in the *IBM CSM for Linux: Administration Guide* to help you diagnose problems before placing a call. If you still need help resolving the problem, call IBM. You might be asked to send relevant data, and to open a problem management record (PMR) for tracking purposes.

## Finding service information

The following Web sites contains all the service bulletins and flashes, as well as PTF and APAR reports for all current releases of CSM:

- Cluster software: <http://techsupport.services.ibm.com/server/support>
- CSM software:  
<https://techsupport.services.ibm.com/server/cluster/fixes/csmfixhome.html>
- CSM for Linux on pSeries:  
[https://techsupport.services.ibm.com/server/cluster/csmplinux\\_1.3.2.0down.html](https://techsupport.services.ibm.com/server/cluster/csmplinux_1.3.2.0down.html)
- CSM for Linux:  
[https://techsupport.services.ibm.com/server/cluster/csmlinux\\_1.3.2.0down.html](https://techsupport.services.ibm.com/server/cluster/csmlinux_1.3.2.0down.html)

## Calling IBM for help

You can get assistance by calling IBM Support. Before you call, be sure you have the following information:

1. Your access code (customer number).
2. The IBM product number. The product number for CSM is 5765-E88.
3. The name and version of the operating system you are using.
4. Any relevant machine type and serial numbers.

5. A telephone number where you can be reached.

The person with whom you speak will ask for the above information and give you a time period during which an IBM representative will call you back.

In the United States:

- The telephone number for IBM software support and IBM hardware support is **1-800-IBM-SERV**.
- The telephone number for IBM Linux support is **1-800-237-5511**.

Outside the United States, contact your local IBM Service Center.

## **Contacting CSM development**

To contact CSM development by e-mail, send your comments to [cluster@us.ibm.com](mailto:cluster@us.ibm.com).





---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM development plans are subject to change or withdrawal without further notice. Any reliance on the Statement of Direction is at the relying party's sole risk and will not create any liability or obligation for IBM.

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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

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 states 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.

IBM may use or distribute any of the information you supply 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 Corporation  
Department LJEB/P905  
2455 South Road Road  
Poughkeepsie, New York 12601-5400  
U.S.A.

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.

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.

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 the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates 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. 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 IBM's application programming interfaces.

---

## Trademarks

The following names are trademarks or registered trademarks in the United States, other countries, or both:

- IBM, alphaWorks, BladeCenter, eServer, the @server logo, IntelliStation, and xSeries are trademarks or registered trademarks of International Business Machines Corp.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

- Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.
- Other company, product and service names may be trademarks or service marks of others.

---

## Publicly Available Software

IBM Cluster Systems Management for Linux includes software that is publicly available:

### **Conserver 7.2 (CSM for Linux on xSeries and pSeries)**

An application that adds logging and multi-user access for remote administration of serial ports, using locally installed multi-port serial interfaces or "reverse-telnet" to console servers, or both.

### **Expect 5.3 (CSM for Linux on xSeries only)**

Tool for automating interactive applications, such as telnet, ftp, passwd, fsck, rlogin, tip.

### **Perl libnet 1.0703 (CSM for Linux on xSeries only)**

Provides a perl client API to FTP, SMTP, NNTP, POP3.

### **SYSLinux 1.64 (CSM for Linux on xSeries only)**

SYSLinux includes PXELINUX, which CSM uses to control the behavior of network boots. SYSLinux is licensed under the GNU GPL.

### **Tftp-HPA 0.34 (CSM for Linux on xSeries and pSeries)**

An implementation of Trivial FTP. Allows download of files from a server when net booting a machine.

### **Rdist 6.15 (CSM for Linux on xSeries only)**

Distributes files to multiple machines, enabling administrators to maintain identical copies of files across multiple machines.

### **Fping 2.42b (CSM for Linux on xSeries Only)**

An implementation of PING for multiple hosts. Quickly ping multiple hosts to determine their reachability.

This book discusses the use of these products only as they apply specifically to the IBM Cluster Systems Management for Linux product.

**Note:** The distribution for SYSLinux includes the source code and associated documentation. All copyright notices and license conditions in the documentation must be respected. You can find version and distribution information for each of these products in the *Specified Operating Environment* section of the *IBM CSM for Linux: Planning and Installation Guide*. For these non-IBM products, the following license terms apply in lieu of the International Program License Agreement.

The freeware package syslinux 1.64 (Freeware Package) is provided with CSM for Linux. The Freeware Package is provided "AS IS" and is not warranted or supported by IBM. IBM expressly disclaims all warranties express or implied as to such Freeware Package INCLUDING THE WARRANTY AGAINST NON-INFRINGEMENT, THE WARRANTY OF MERCHANTABILITY AND THE WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE. IBM further disclaims

all liability for any damages (including without limitation direct and indirect damages) arising in connection with the Freeware Package. The Freeware Package is licensed under the terms of the GPL, a copy of which is included in the Freeware Package.

Open Source Software	Download site:
Autoupdate 4.3.4 or higher (only required for software maintenance: installation and upgrades of non-CSM RPM packages) (CSM for Linux on xSeries and pSeries)	<a href="http://freshmeat.net/projects/autoupdate">http://freshmeat.net/projects/autoupdate</a>
perl-XML-Parser (CSM for Linux on xSeries)	SuSE SLES 7: <a href="ftp://ftp.rpmfind.net/linux/SuSE-Linux/i386/7.2/full-names/i386/perl-XML-Parser-2.27-63.i386.rpm">ftp://ftp.rpmfind.net/linux/SuSE-Linux/i386/7.2/full-names/i386/perl-XML-Parser-2.27-63.i386.rpm</a>
perl-XML-Simple (CSM for Linux on xSeries)	<ul style="list-style-type: none"> <li>• SuSE SLES 7: <a href="ftp://speakeasy.rpmfind.net/linux/Mandrake-devel/cookfire/i586/Mandrake/RPMS/perl-XML-Simple-1.05-1mdk.i586.rpm">ftp://speakeasy.rpmfind.net/linux/Mandrake-devel/cookfire/i586/Mandrake/RPMS/perl-XML-Simple-1.05-1mdk.i586.rpm</a></li> <li>• SuSE SLES 8: <a href="ftp://ftp.suse.com/pub/suse/i386/8.1/suse/i586/perl-XML-Simple-1.08-43.i586.rpm">ftp://ftp.suse.com/pub/suse/i386/8.1/suse/i586/perl-XML-Simple-1.08-43.i586.rpm</a></li> </ul>

The following non-IBM software is required if you want to perform remote hardware control operations for IBM pSeries servers attached with a Hardware Management Console (HMC):

- openCIMOM Version 0.7

You can download the software from:

- <https://techsupport.services.ibm.com/server/cluster>.

The following non-IBM software is required for CSM for Linux on pSeries is required if you want to perform the software maintenance installation and upgrade of non-CSM RPMs on Linux Managed nodes from the management server:

- Autoupdate V4.8, or later levels which maintain full backward compatibility.

You can download the software from:

- <http://freshmeat.net/projects/autoupdate>.

In addition to the terms above, certain components of the Program have Program-unique terms, which are identified in a "read me" file in the Program.

**EXCLUDED COMPONENTS:** Notwithstanding the terms and conditions of any other agreement you may have with IBM or any of its related or affiliated companies (collectively "IBM"), the following terms and conditions apply to all "Excluded Components" identified in this document: (a) all Excluded Components are provided on an "AS IS" basis; (b) IBM DISCLAIMS ANY AND ALL EXPRESS AND IMPLIED WARRANTIES AND CONDITIONS INCLUDING, BUT NOT LIMITED TO, THE WARRANTY OF NON-INFRINGEMENT OR INTERFERENCE AND THE IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE; (c) IBM will not be liable to you or indemnify you for any claims related to the Excluded Components; and (d) IBM will not be liable for any direct, indirect, incidental, special exemplary, punitive or consequential damages with respect to the Excluded Components.

The following components in the Program are Excluded Components: (a) conserver 7.2, (b) perl-libnet 1.0703, (c) expect 5.3, (d) perl-to-c extensions and (e) fping 2.4b2, (f) rdist 6.15, and (e) tftp-HPA 0.34.

CSM for Linux includes software developed by the Ohio State University and its contributors.

The inclusion herein of copies of various licenses is not meant to imply endorsement of the principles, methodologies, or views that are contained therein, either express or implied.

For SYSLinux code, licensed under the GNU GENERAL PUBLIC LICENSE Version 2, June 1991, see the following.

Copyright (c) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### **Preamble**

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## **GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program. You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
  - a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
  - b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
  - c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.



Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
  - a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
  - b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
  - c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You

may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances. It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE



COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS



---

# Index

## A

about this book v  
addlcsext command 2  
audience of this book v

## B

Bibliography 157

## C

calling IBM 158  
cfmupdatenode command 5  
chnode command 9  
chrconsolecfg command 11  
chsnmp command 14  
commands  
  addlcsext 2  
  cfmupdatenode 5  
  chnode 9  
  chrconsolecfg 11  
  chsnmp 14  
  copycsmpkgs 17  
  CSM 1  
  csmbackup 22  
  csmconfig 25  
  csmrestore 28  
  csmsetupks 30  
  csmsetupyast 35  
  csmsis2yast 39  
  csmstat 41  
  dcem 44  
  definenode 46  
  dsh 54  
  dshbak 62  
  getadapters 64  
  installms 68  
  installnode 71  
  lshwinfo 78  
  lshwstat 81  
  lsnode 84  
  lssnmp 87  
  mgmtsvr 90  
  monitorinstall 92  
  nodegrp 102  
  predefined-condresp 110  
  predefined-nodegroups 112  
  probemgr 114  
  rconsole 118  
  rconsolerefresh 122  
  reventlog 124  
  rmlcsext 126  
  rmnode 128  
  rpower 130  
  smsupdatenode 134  
  systemid 139  
  uninstallms 141

commands (*continued*)  
  updateisvr 143  
  updatenode 145  
conventions v  
copycsmpkgs command 17  
CSM commands 1  
  csm commands, list 19  
  CSM information 157  
  csm informational man page 19  
  CSM project e-mail address 159  
csmbackup command 22  
csmconfig command 25  
csmrestore command 28  
csmsetupks command 30  
csmsetupyast command 35  
csmsis2yast command 39  
csmstat command 41

## D

dcem command 44  
definenode command 46  
dsh command 54  
dshbak 62

## E

e-mail address for CSM 158, 159

## F

files  
  kscfg.tmpl 76  
  nodeattributes 95  
  nodedef 100

## G

getadapters command 64  
glossary 151

## H

help from IBM 158

## I

informational man pages  
  csm 19  
installms command 68  
installnode command 71  
ISO 9000 v

## K

kscfg.tmpl file 76

## L

log files 149  
lshwinfo command 78  
lshwstat command 81  
lsnode command 84  
lssnmp command 87

## M

mailing list, CSM 158  
mgmtsvr command 90  
monitorinstall command 92

## N

node definition file (see nodedef file) 100  
nodeattributes 95  
nodedef file 100  
nodegrp command 102  
noderange 105

## P

predefined-condresp command 110  
predefined-nodegroups command 112  
prerequisite knowledge for this book v  
probemgr command 114  
publications  
    obtaining 157  
    related 157  
publicly available software 163

## R

rconsole command 118  
rconsolerefresh command 122  
registered trademarks 162  
related information 157  
reventlog command 124  
rmlcsex command 126  
rmnode command 128  
rpower command 130

## S

smsupdatenode command 134  
systemid command 139

## T

trademarks 162

## U

uninstallms command 141  
updateisvr command 143  
updatenode command 145

## W

Web sites for publications 157

## X

XCAT tools 158

---

# Readers' Comments — We'd Like to Hear from You

IBM Cluster Systems Management for Linux  
Command and Technical Reference  
Version 1.3.2

Publication No. SA22-7933-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  Yes  No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

---

Name

---

Address

---

Company or Organization

---

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation  
Department 55JA, Mail Station P384  
2455 South Road  
Poughkeepsie NY 12601-5400



Fold and Tape

Please do not staple

Fold and Tape





Program Number: 5765-E88, 5765-G16

SA22-7933-01

