IBM Cluster Systems Management for Linux®

Technical Reference

Version 1.1

SA22-7851-02

IBM Cluster Systems Management for Linux®

Technical Reference

Version 1.1

SA22-7851-02

Note!

Before using this information and the product it supports, read the information in “Notices” on page 175.

Third Edition (December 2001)

This edition of the *IBM Cluster Systems Management for Linux Technical Reference* applies to IBM Cluster Systems Management for Linux Version 1.1, program number 5765–E88, and to all subsequent releases of this product until otherwise indicated in new editions.

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: mhvrcfs@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.

Contents

About This Book

This book is the updated version of the *IBM Cluster Systems Management for Linux Technical Reference* (SA22-7851-00) that was last published in June, 2001. This book describes the commands and files that are used to implement the IBM Cluster Systems Management for Linux suite of tools.

Who Should Use This Book

This book is intended for system administrators who want to use the command-line interface for IBM Cluster Systems Management for Linux. The system administrator should be experienced with UNIX® and networked systems.

How to Use This Book

This book contains information that describes the IBM Cluster Systems Management for Linux command-line interface. It is divided into four sections.

1. The first section contains commands that are exclusively for the IBM Cluster Systems Management for Linux command-line interface.
2. The second section contains commands that implement conditions and responses for the Monitoring application.
3. The third section contains commands that interact with the Resource Monitoring and Control (RMC) subsystem.
4. The last section contains scripts and utilities. The scripts can be used as-is or modified to create user-defined scripts that are tailored for your installation.

Highlighting

The following highlighting conventions are used in this book:

Bold	Identifies commands, subroutines, keywords, files, structures, directories, and other items whose names are predefined by the system. Also identifies graphical objects such as buttons, labels, and icons that the user selects.
Italic	Identifies parameters whose actual names or values are to be supplied by the user.
`monospace`	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

ISO 9000

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

Related Information

The following references contain more information about IBM Cluster Systems Management for Linux:

1. *IBM Cluster Systems Management for Linux Administration Guide*, SA22-7873-00
2. *IBM Cluster Systems Management for Linux Planning and Installation Guide*, SA22-7853-01
3. *IBM Cluster Systems Management for Linux Remote Control Guide and Reference*, SA22–7856-01

How to Obtain Publications

The IBM Cluster Systems Management for Linux publications are available as HTML and PDF files on the CD-ROM in the **/doc** directory or on the installed system in the **/opt/csm/doc** directory. The README is available on the CD-ROM in the root directory (/). The file names are as follows:

- *IBM Cluster Systems Management for Linux Administration Guide*, am7LXadm.pdf
- *IBM Cluster Systems Management for Linux Planning and Installation Guide*, am7LXstp.pdf
- *IBM Cluster Systems Management for Linux Remote Control Guide and Reference*, am7LXrem.pdf
- *IBM Cluster Systems Management for Linux Technical Reference*, am7LXcmd.pdf

Publications for IBM Cluster Systems Management for Linux were available also at the time of this release at the following Web site:

`http://www.ibm.com/eserver/clusters/linux`

How to Reach Us by E-Mail

If you would like to contact us by e-mail, send your comments to cluster@us.ibm.com

Chapter 1. CSM Commands

addnode Command

Name

addnode - Defines and installs nodes in a cluster.

Synopsis

addnode [**−h**]

addnode [**−v**] **−f** *nodedef* [*Attr=value* [*Attr=value* ...]]

addnode [**−v**] [**−n** *starting_node*] [**−c** *count*] [**−H** *HWControlPoints*] [**−C** *ConsoleServers*] [**−t** *HWtype*][**−x |** **−p** *pkg_path*] [**−m** *install_method*] [*Attr=value* [*Attr=value* ...]]

Description

The **addnode** command defines and installs the managed nodes in a cluster. Before running this command, you must run the **installms** command to install the management server. The **addnode** command runs on the management server synchronously for a CSM-only installation and asynchronously for a full installation. This command can be rerun. If you do not provide some of the arguments, the program prompts you for each piece of information that it needs.

When the **addnode** command is run for a CSM-only installation and it is completed successfully, the cluster is ready for use. When the **addnode** command is run for a full installation, it initiates the installation process and exits usually before the installation is completed.

Before running a full installation, set the node boot order to the following:

- Floppy
- CD-ROM
- Network
- Hard disk

To install CSM only, **addnode** does the following:

1. Runs the **definenode** command to define the nodes in the cluster.
2. Runs **installnode** to do the following:
   a. Performs an **NFS** mount of **/tftpboot** to each node by means of **dsh**.
   b. Runs **makenode** from the mounted **/tftpboot** on each node by using **dsh.**

If you want to verify and perhaps change node definitions before you complete a CSM-only installation, you need to run the underlying commands separately.

To do a full installation, **addnode** does the following:

1. Runs the **definenode** command to define the nodes in the cluster.
2. Runs the **setupks** command to set up Kickstart.
3. Runs **installnode**.

   When the **installnode** command is run for a full installation, there are arguments for which **addnode** provides the default values for a standard installation. If you need to customize these arguments, instead of running **addnode**, you need to run the underlying commands separately.

Attributes matching the **setupks** command attributes can be specified on the command line by attribute/value pairs 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:

- Netmask

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

See “setupks Command” on page 74 for more information.

The **addnode** command generates the **/etc/opt/csm/netfinity_power.config** file. A template file is shipped in **csm.server**. See the **/etc/opt/csm/netfinity_power.config.tmpl**. This template file contains a default user ID and password, which is supplied for the service processor when it is shipped from the factory. The **addnode** command uses the default user ID and password from the template file to generate the user ID and password for all of the nodes in the cluster. To change the password or user ID for a node from the default, the file created from the template, **netfinity_power.config**, when **addnode** is run, must be edited manually after **addnode** is run. Additional steps for changing these user IDs and passwords are supplied in the *IBM Cluster Systems Management for Linux Remote Control Guide and Reference*.

Notes:

1. You cannot redefine a node that is already defined. If you attempt to define a node that is already defined by using the **−n** option, the **addnode** command fails and returns an error message stating that the node is already defined. The command can be rerun and the defined node can be avoided when the command is rerun. But, if a defined node is listed in the node definition file (*nodedef*), processing is completed for the nodes that are undefined, and an error message is generated for the already defined nodes.

   Because nodes cannot be redefined, **addnode** can only be used for an initial installation, not an update. To do an update the **setupks** command (for a full installation) and the **installnode** command must be run again.
2. The **addnode** command does not add the host names to the nameserver or to the **/etc/hosts** file on the management server. You must add the host name to the nameserver or to **/etc/hosts** manually before **addnode** is run.

The **addnode** command creates ManagedNode objects based upon user input in the CSM database (in the ManagedNode class) for each node and installs each node. See the **lsnode** man page for details about the attributes that are added to the CSM database.

To change the attributes of a node, use the **chnode** command.

Options

−c *count*
  Specifies how many nodes to define. IP addresses are calculated for nodes beyond the IP address for the starting node. The default is one.

−C *ConsoleServers*
  Specifies the list of console server definitions. Multiple console server definitions are separated by commas. To specify that there are no console servers, enter a null value as follows:

```
-C " "
```

Note: A blank space is required between the opening and closing double quotation marks.

Each entry is formatted as follows:

```
-C csname[:method:csnum:starting_portnum][, ...]
```

where:

csname
  Represents the host name or IP address of a console server.

method
  Represents the console method. The default method is **esp**. Valid values for this field are

`esp`, `conserver`, or `none`, or the field can be left blank. If the console server name field is left blank (the **-C** option has a value of ″ ″), then the value assumed for this field is `none`. If the console server name field is not blank and this field is blank, the default value of `esp` is assumed.

csnum Represents the console server number. The default is 1.

starting_portnum
Represents the starting console port number. The default is 0. If some ports have already been defined for the console server, then a starting port number higher than 0 needs to be used. The console port number is represented by a single hexadecimal character (0–f). The number of ports available to each console server depends on the console method.

If the console method is the default **esp**, the program determines how many ports are available. If you would like to know this information, type:

```
esptty -c
```

Otherwise, the default is 16. The port number is incremented up to the number of nodes to be defined. Each node is assigned a specific port number and console server. All the ports for a console server are assigned, and then the ports for the next console server are assigned. The console servers are used in the order that they are specified with the **-C** option.

If any of the port numbers for a console server are already in use by an existing ManagedNode or PreManagedNode, the **addnode** command fails without defining any nodes.

−f *nodedef*
Represents the node definition file. This file contains the node list, the console_server definitions, and the hardware control point definitions. A sample node definition file is supplied in **/opt/csm/install/nodedef.sample**. This option cannot be used with any other option except **−v** and *Attr=value*.

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

−H *HWControlPoints*
Specifies the list of hardware control points and their associated service processors. Multiple hardware control point definitions are separated by commas. To specify that there are no hardware control points, enter a null value as follows:

```
-H " "
```

Note: A blank space is required between the opening and closing double quotation marks.

Each hardware control point definition for a node is specified in the following format:

```
-H HWCtrlPt[:method:service_processor_name][,... ]
```

where:

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

method
Represents the power method. Valid values are `netfinity` or `none`, or the field can be left blank. If the hardware control point is listed and the hardware type is `netfinity`, the default hardware type is `netfinity`. If the hardware control point field is not listed (the **-H** option has a value of ″ ″) or the hardware type is not `netfinity`, then the value assumed for this field is "`none`".

service_processor_name
Represents the starting internal service processor (ISP) name. A maximum of 10 service processors are connected to each hardware control point. If a starting service processor

name is not provided, the default is **node01**. There are two formats provided for the service processor name. The first format is **node***nn*, where *nn* is an integer from 01 through 10. A starting service processor higher than **node01** needs to be used if some of the service processors have already been defined for a hardware control point. The second format is based on the short host name of each node and is specified by entering the word **hostname** as the value for *service_processor_name*.

Note: This changes the setting of the CSM database **SvcProcName** attribute name of the ISP to the short host name of the node. To change the actual ISP name (text id) to the short host name of the node, refer to the *IBM Cluster Systems Management for Linux Remote Control Guide and Reference*. It is worthwhile to change the ISP to the short host name, for example, if you want to track SNMP (Simple Network Management Protocol) alerts. The ISP provides reporting mechanism support for tracking SNMP alerts.

The service processor name can be incremented up to the number of nodes to be defined. Each node is assigned to a specific service processor and hardware control point. All the service processors attached to a hardware control point are assigned, and then the service processors attached to the next hardware control point are assigned. The hardware control points are used in the order that they are specified with the **-H** option.

If any of the hardware control points for a management service processor are already in use by an existing ManagedNode or PreManagedNode, then the **addnode** command fails without defining any nodes.

−m *install_method*
Specifies the installation method. There are two valid values, **csmonly** (to install just CSM) and **kickstart** (to install the operating system and CSM). The default is **csmonly**.

−n *starting_node*
Specifies the IP address or host name of the starting node. A list of nodes can be generated by incrementing IP addresses up to the value specified by ***count***. If a host name is specified, it is converted to an IP address.

−p *pkg_path*
Specifies one or more directories, separated by colons (for example, *path1:path2:path3*), that contain copies of the Red Hat CD-ROMs. The default is **/mnt/cdrom**. This is passed to **setupks**.

−t *HWType*
Specifies the hardware type for all of the nodes to be defined. If the hardware type is set to `netfinity`, then PowerMethod defaults to `netfinity`, provided the hardware control point is set.

−v Writes the command’s verbose messages to standard output.

−x Specifies to not copy Red Hat disks. This cannot be used with the **-p** option.

Examples

1. To be prompted for all the necessary information, type:
```
addnode
```
The program requests the required information from you.
2. To add one node to an existing cluster, type:
```
addnode -nc175n11.ppd
```
3. To add 16 nodes to a new cluster, type:
```
addnode -ncls02 -c16 -Hmgt03,mgt04 -Cmgt01 -tnetfinity
```
4. To add four nodes to an existing cluster, type:
```
addnode -ncls18 -c4 -Hmgt04::node07 -Cmgtn02 -tnetfinity
```

5. To define five nodes and their associated hardware control points and console servers, with a hardware type of `netfinity`, type:

   ```
   addnode -nclsn02 -c5 -Hmgtn03 -Cmgtn02 -tnetfinity
   ```

For a CSM-only install, output from the **addnode** command is similar to:

```
# addnode -nclsn05 -Hmgtn03::node05 -Cmgtn02:::4 -mcsmonly -tnetfinity
definenode: Adding CSM Nodes:
definenode: Adding Node clsn05.ppd.pok.ibm.com(9.114.133.197)
installnode: Output log is being written to /var/log/csm/installnode.log
installnode: Nodes to full install via kickstart:

installnode: Nodes to CSM-Only install via dsh:
        clsn05.ppd.pok.ibm.com
installnode: Installing CSM-only on nodes:
        clsn05.ppd.pok.ibm.com
clsn05.ppd.pok.ibm.com: umount: /tftpboot: not mounted
installnode: clsn05.ppd.pok.ibm.com CSM-only install SUCCESSFUL!
installnode: Status of nodes after the install:
Node                           Status
---------------------------------------------------------
c5bn02.ppd.pok.ibm.com         Installed
c5bn03.ppd.pok.ibm.com         Installed
c5bn04.ppd.pok.ibm.com         Installed
c5bn05.ppd.pok.ibm.com         Installed

installnode: Run /opt/csm/bin/monitorinstall to display install status
```

For a full install of CSM and Linux, output from the **addnode** command is similar to:

```
# addnode -n clsn06.ppd.pok.ibm.com -c1
Enter list of Hardware Control Points (press ENTER for none):
        Format:  hwctrlpt[:method:spname][,...]
                 mspname = Hardware Control Point hostname
                           or IP address.
                 method  = Power method (default=netfinity)
                 spname  = Starting service processor name
                           or 'hostname' (default=node01)
        Example: hwctrlpt1::node06,hwctrlpt2,hwctrlpt3
        Example: hwctrlpt1::hostname,hwctrlpt2::hostname
mgtn03::node06
Enter list of Console Servers (press ENTER for none):
        Format: csname[:method:csnum:port][, ...]
                csname = Console server name (hostname or IP address)
                method = Console method (default=esp)
                csnum  = Console server number (default=1)
                port   = Starting console port number (default=0)
        Example: cs1:::4,cs2:conserver,cs3
mgtn02:::5
Enter Hardware Type (default = netfinity): netfinity
Enter Install Method (csmonly or kickstart, default = csmonly):  kickstart
definenode: Adding CSM Nodes:
definenode: Adding Node clsn06.ppd.pok.ibm.com(9.114.133.206)
setupks: Copying RedHat Images from /mnt/cdrom
setupks: Insert Red Hat Linux 7.1 disk 1.
Press Enter to continue
setupks: Insert Red Hat Linux 7.1 disk 2.
Press Enter to continue
8192+0 records in8192+0 records outmke2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95
/08/098192+0 records in8192+0 records out 62.6%clsn06.ppd.pok.ibm.com: 00:03:47:4d
:44:4a 9.114.133.206
installnode: Output log is being written to /var/log/csm/installnode.log
installnode: Nodes to full install via kickstart:
        clsn06.ppd.pok.ibm.com
installnode: Nodes to CSM-Only install via dsh:

installnode: Rebooting Node for full install: clsn06.ppd.pok.ibm.com
```

```
installnode: Status of nodes after the install:
Node                           Status
---------------------------------------------------------
clsn02.ppd.pok.ibm.com         Installed
clsn03.ppd.pok.ibm.com         Installed
clsn04.ppd.pok.ibm.com         Installed
clsn05.ppd.pok.ibm.com         Installed
clsn06.ppd.pok.ibm.com         Rebooting to Install Node

installnode: Run /opt/csm/bin/monitorinstall to display install status
```

Files

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

/opt/csm/install/kscfg.tmpl
Location of the Kickstart configuration file template.

/opt/csm/install/firstboot.tmpl
Location of the first boot file template.

/etc/opt/csm/netfinity_power.config.tmpl
Location of the template file that contains a default user ID and password, which is supplied for the service processor when it is shipped from the factory. The **addnode** command uses the default user ID and password from the template file to create user IDs and passwords for all the nodes in the cluster. The **addnode** command generates the **netfinity_power.config** file to contain these passwords.

/etc/opt/csm/netfinity_power.config
Location of the service processor password file. To change the user ID and password for a node, this file must be edited manually after **addnode** is run.

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

See Also

- The **definenode**, **installms**, **installnode**, **makenode**, **monitorinstall**, **rconsole**, **rpower**, **setupks** commands.
- The **kscfg.tmpl** file, **netfinity_power.config** file, **netfinity_power.config.tmpl** file, **nodedef** file
- The **rmccli General Information** file for information on attr=value syntax
- *IBM Cluster Systems Management for Linux Planning and Installation Guide* (am7LXstp.pdf)
- *IBM Cluster Systems Management for Linux Remote Control Guide and Reference* (am7LXrem.pdf)

Author

Sean Safron - cluster@us.ibm.com

cfm Command

Name

cfm - Starts the configure file manager process, which copies or provides links to all new or modified files in the server's file repository.

Synopsis

cfm [**−h**]

Description

The **cfm** command provides the major function of the configuration file manager application. The configuration file manager (CFM) provides a directory or file system that contains all the files that are shared among CFM managed nodes. The system administrator builds this directory to be the repository of files or links to files of interest, in particular, configuration files. The repository is set up on the management server, and an agent pulls changes to the files in the repository down to the managed nodes in the cluster. For a specified configuration file, versions can be set up for all nodes, for particular types of nodes, or for a specific host name.

Note: If symbolic links are used in the repository, the links and not the files that they point to are copied over. Do not use symbolic links to distribute the actual configuration files themselves.

The **cfm** command is usually run on the managed node. A managed node can be instructed to pull new or changed configuration files at boot time, when the files change, and when the administrator runs a command (such as **cforce**) on the managed node.

The top of the repository is **/cfmroot**. This directory should be considered to be equivalent to the root directory (**/**) on the managed nodes (for example, **/cfmroot/etc** on the server equates to **/etc** on the managed node). Files can be copied or linked, but copying allows you to do variable substitutions that are not possible by linking.

To do variable substitution, the **cfm** command checks the files in the file repository of the server and copies all new or modified files. When the variables *%ip%* and *%hostname%* are found in a configuration file, the **cfm** command substitutes the IP address and host name of the managed node on which **cfm** is running for these variables. See example 3 on page 9 for an illustration of how this works.

In addition, **cfm** allows preprocessing and postprocessing; for example, to lock files that are being copied. If a *filename***.pre** or *filename***.post** exists, before the configuration file is copied, the *filename***.pre** script or executable is run, and after the configuration file is copied, *filename***.post** is run.

To copy variations of the same file for different nodes, CFM uses the concept of groups as defined by the CSM **nodegrp** command. To associate a file with a specific group, add a **._***GroupName* extension to the file name. For example, **inetd.conf._aixbin** copies files only to nodes in the **aixbin** group. To see a list of all the defined node groups in the cluster, type:

```
nodegrp -l
```

Check **cfengine.log** for reports of any problems during set up or during file synchronization. This log contains both the standard error and the standard output messages from cfengine. It is currently located in **/var/log/cfengine.log**. The entries include all files changed on the local node in user-readable format. Because the log is not circular, but rather is appended upon each run of **cfm**, the administrator needs to prune the log entries periodically according to the system policy for that installation.

The **cfm** command uses the file copy capability of the cfengine GNU tool.

Notes:

1. Normally **cfm** is not issued at the command line; rather it is called by the internal **cfd** mechanism on a regular basis (similar to a **cron** job). The **cfm** command can also be called by the **cforce** command. In addition, a predefined condition (CFMRootModeTimeChanged) can be set up and associated with **cforce** as a response action. This forces file synchronization to occur if one or more files change on the management server. (See the example section.)
2. **cfm** is sensitive to time differences between the management server and the client nodes. It is the administrator's responsibility to ensure that time is kept in synchronization by means of **ntp** or another suitable method.
3. **cfm** queries node group definitions during processing on each node. This implies that the **mgmtsvr** setting on each node must be set correctly and that the RMC ACL file on the management server node must give access to each node. These are normally set up by CSM automatically. To ensure that the proper permissions are in place, run the following command on each node:

   ```
   nodegrp -l
   ```

   If you can see the defined node groups, then CFM is able to process node group extensions to file names correctly.

Options

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

Examples

1. To run the configuration file manager on the current managed node, type:

   ```
   cfm
   ```

2. To copy a file into **/cfmroot** from **/etc**, keeping the same permissions, ownership, modification, and access times, type:

   ```
   cp -p /etc/inetd.conf /cfmroot/etc/inetd.conf
   ```

   The output is similar to:

   ```
   /cfmroot/etc/inetd.conf
   /cfmroot/usr/home/fred/tune.cnf
   /cfmroot/etc/hosts
   ```

   The output on the managed node system after the **cfm** command distributes the files from the server is similar to:

   ```
   /etc/inetd.conf
   /usr/home/fred/tune.cnf
   /etc/hosts
   ```

3. To use variable substitution in order to add the node with an IP address of 192.168.100.2 and a host name of **lister** to the **/etc/hosts** file, add the *%ip%* and *%hostname%* variables to the **/cfmroot/etc/hosts** file as follows:

   ```
   192.168.100.4   clifford
   192.168.100.5 snowy
   192.168.100.6 ns1
   192.168.100.20 streika
   192.168.100.21 bars
   192.168.100.22 lisichka
   %ip% %hostname%
   ```

   On the managed node, this resolves to:

   ```
   192.168.100.4   clifford
   192.168.100.5 snowy
   192.168.100.6 ns1
   ```

```
192.168.100.20 streika
192.168.100.21 bars
192.168.100.22 lisichka
192.168.100.2 lister
```

4. To force file synchronization to occur if one or more files change on the management server by using the monitoring application, type:

   ```
   startcondresp CFMRootModTimeChanged Cforce
   ```

Diagnostics

All standard error output is displayed and also recorded in the log file.

Files

These files should not be modified:

/usr/local/sbin/cfengine
Location of the cfengine program

/etc/opt/csm/cf.copy
Location of the cfengine script for copying files from the server repository to the local cache.

/etc/opt/csm/cf.copylocal
Location of the cfengine script for copying files from the local cache to the final destination.

/etc/opt/csm/cf.main
Location of the generic cfengine configuration file that contains items such as the cfengine server. It also serves as the security file for the managed node; for example, it lists the servers that the managed node can access.

/etc/opt/csm/cfd.conf
Location of the configuration file for the **cfd** daemon portion of cfengine. This file serves as the resident security file for the server and determines the nodes that can access this server.

/etc/opt/csm/cfengine.conf
Location of the primary configuration file for **cfm**. It handles the copying of files from the server repository to a local cache.

/etc/opt/csm/cfengine.conf.local
Location of the secondary configuration file for **cfm**. It handles the copying of files from the local cache to the final destination.

/var/log/cfengine.log
Location of the cfengine log file.

See Also

The **cfengine**, **cforce**, **nodegrp** commands.

IBM Cluster Systems Management for Linux Administration Guide for information on the RMC ACL file and predefined conditions and responses.

Author

Christopher Hawkinson- cluster@us.ibm.com

cforce Command

Name

cforce - Forces the **cfm** managed node to be run on a specific system for file synchronization.

Synopsis

cforce [−h] [*host*]

Description

The **cforce** command is run on the management server and causes all the nodes to pull down any new configuration files from the management server. The **cforce** can optionally cause **cfm** to be run on a single node when the optional parameter *host_name* is specified.

host	Specifies the node on which to run **cfm**.

Options

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

Examples

1. To force **cfm** to run on all nodes, enter:
   ```
   cforce
   ```
2. To run **cfm** on one particular managed node, in this case, a node with the host name of **puppy**, enter:
   ```
   cforce puppy
   ```

Files

/etc/opt/csm/cfd.conf	Location of the configuration file for the **cfd** daemon portion of cfengine.

See Also

The **cfengine**, **cfm** commands.

Author

Christopher Hawkinson - cluster@us.ibm.com

chnode Command

Name

chnode - Changes a node definition in the IBM Cluster Systems Management for Linux (CSM) database.

Synopsis

chnode [**−h**] [**−P**] [**−v**] {*host* **|** **−w** *selectstr*} *Attr=value* [*Attr=value* ...]

Description

The **chnode** command changes a managed 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/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 *host* parameter can be specified by host name or by IP address. If **-w** *selectstr* is specified, **chnode** uses that 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 on any node, including the management server, to change a node definition in the IBM Cluster Systems Management for Linux database on the management server.

Options

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

−P Changes nodes in the PreManagedNode table/class. (Normally, nodes are changed in the managedNode table/class.)

−v Writes the command’s verbose messages to standard output.

−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). As a convenience, ″*″ means all nodes, as if a ″where″ string were not specified. If **whichdb** is set to **rmc**, then the syntax for the select string is described in the section ″Using Expressions″ in the *IBM Cluster Systems Management for Linux Administration Guide*. If **whichdb** is set to **file**, then normal SQL syntax is used.

Exit Status

1 An invalid combination of options and arguments has been entered.

12 Node not found.

51 Badly formed select string. This error is returned from the DBI layer.

When an error is detected in the DBI layer, the return code of the DBI layer is used as the exit status. When RMC is used as the DBI layer, the exit status of the RMC command is returned as the exit status. The DBI (DataBaseInterface) is a perl interface to databases that is similar in purpose to ODBC. DBI drivers (DBDs) are available for many kinds of databases.

Examples

1. To change the operating-system version for **websvr** in the CSM database, type:

```
chnode websvr OSversion='7.1'
```

Files

/opt/csm/bin/chnode	Location of the **chnode** command

See Also

The **createnode**, **lsnode**, **nodegrp**, **rmnode**, **whichdb** commands.

The **rmccli General Information** file for information on attr=value syntax.

IBM Cluster Systems Management for Linux Administration Guide for information on selection string syntax.

Author

Bruce Potter - cluster@us.ibm.com

chsensor Command

Name

chsensor - Changes attributes of an RMC sensor.

Synopsis

chsensor [**−h**] [**−v | V**] [**−n** *nodelist*] *Name* [*Attr=value* [*Attr=value*...]]

Description

The **chsensor** command changes one or more attributes of the sensor identified by *Name*. Currently, the only attribute that can be changed after the definition of the sensor is its Name; for example, the RefreshInterval attribute cannot be changed. If the value is a string that contains spaces or other special characters, the value must be enclosed in quotation marks. This command returns the error messages and exit status of the underlying **chrsrc** RMC command.

The **chsensor** command runs on any node, including the management server, and normally operates on the local node where it is run unless CT_CONTACT is set.

See the **mksensor** command for a more complete explanation of RMC sensors.

Options

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

−n *nodelist*
When in high-availability (HA) cluster mode, this option specifies the node numbers for the nodes on which this sensor is to be changed. To change the sensor on the local node only, set the CT_LOCAL_SCOPE environment variable to one before running this command. This option is not necessary when not in HA cluster mode.

−v Writes the command’s verbose messages to standard output. This option can be entered either uppercase or lowercase.

−V Writes the command’s verbose messages to standard output. This option can be entered either uppercase or lowercase.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred with RMC that was based on faulty command line input.

6	No resources found.

Security

The user needs write permission for the IBM.Sensor resource class in order to run **chsensor**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To change the name of the NumLogins sensor to Numlog, type:

   ```
   chsensor NumLogins Name=NumLog
   ```

Files

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

See Also

The **chrsrc**, **lssensor**, **mksensor**, **rmsensor** commands.

The **rmccli General Information** file for information on attr=value syntax.

The *IBM Cluster Systems Management for Linux Administration Guide* for information on the ACL authorization file and on selection string syntax.

Author

Bruce Potter - cluster@us.ibm.com

createnode Command

Name

createnode - Creates a node definition in the IBM Cluster Systems Management for Linux (CSM) database.

Synopsis

createnode [**−h**] [−M] [**−v**] *host* [*Attr=value [Attr=value* ...]]

Description

This command is normally not run from the command line. Instead **createnode** is called as part of the automated installation process. This command is run on the management server.

The **createnode** command adds a managed-node definition to the CSM database as an additional node to be managed by this management server. The Hostname attribute in the database is set by looking up the primary DNS host name of the specified *host*. The *host* parameter can be specified by host name or by IP address.

Additional attributes can be set in the database for this node definition 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.

Attribute values can be changed later with the **chnode** command.

Options

−h	Writes the command’s usage statement to standard output.
−M	Adds this node to the ManagedNode table/class. Normally, the node is added to the PreManagedNode class and moved to the ManagedNode class later when the CSM installation code has set up the node.
−v	Writes the command’s verbose messages to standard output.

Exit Status

1	An invalid combination of options and arguments has been entered.
12	Node already exists.
51	Badly formed select string. This error is returned from the DBI layer.

When an error is detected in the DBI layer, the return code of the DBI layer is used as the exit status. When the Resource Monitoring and Control subsystem (RMC) is used as the DBI layer, the exit status of the RMC command is returned as the exit status. The DBI (DataBaseInterface) is a perl interface to databases that is similar in purpose to ODBC. DBI drivers (DBDs) are available for many kinds of databases.

Examples

1. To create a node **websvr** in the CSM database, type:

```
createnode websvr HWtype='netfinity'
```

Files

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

See Also

The **chnode**, **lsnode**, **nodegrp**, **rmnode**, **whichdb** commands.

The **rmccli General Information** file for information on attr=value syntax.

Author

Bruce Potter - cluster@us.ibm.com

dcem Command

Name

dcem - Provides a graphical user interface (GUI) that allows you to run a command or script on multiple distributed nodes on a network at the same time.

Synopsis

dcem [**−h | −−help**] [**−V | −−version**] [**−v | −−verbose**] [**−N | −−group** [*group,group,group,...*]] [**−n | −−hosts** [*host_name,host_name,host_name,...*]] [*command_specification_name*]

Description

Distributed Command Execution Manager (DCEM) provides a GUI that allows you to run a command or script on multiple distributed machines on a network at the same time. You can specify a collection of individual nodes, or you can create groups of nodes and save them to use again. DCEM provides real-time command execution status on the individual nodes that you specify, showing them in a waiting, working, successful, or failed state. It helps you to create, save, and edit command specifications, and it also creates a log of all distributed command activity. DCEM provides the capability of saving command specifications as PERL scripts. These saved command specifications can be executed at the command line. The command specifications are saved under the user's home directory **/home/dcem/scripts**.

The *command_specification_name* parameter initializes the input fields with the specified command. This is the name that is used when the command is saved in the DCEM GUI.

For the first release, the **dsh** command provides the underlying function.

Options

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

−V | −−version
Writes version information to standard output.

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

−N | −−groups [*group, group,group,...***]**
Specifies the name of a group of hosts displayed in the **Groups of hosts** field in the DCEM dialog. If you use this option with the command specification name, the host names and groups that were saved with the command are ignored.

−n | −−hosts [*host_name,host_name,host_name,...***]**
Specifies the name of the hosts displayed in the **Host names** field of the DCEM dialog. If you use this option with the command specification name, the host names and groups that were saved with the command are ignored.

Examples

The following are examples for specifying 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, type:
   ```
   dcem
   ```
2. To initialize the input fields with specified command name and groups, type:
   ```
   dcem --groups 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, type:

   ```
   dcem --groups 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 version of DCEM that is running, type:

   ```
   dcem -V
   ```

Files

/home/dcem/scripts/*script_filename***.pl**	Location of the DCEM command specification scripts
/home/dcem/logs/*log_filename*	Location of DCEM log files

See Also

The **dsh** command

Author

Sandy Amin - cluster@us.ibm.com

definenode Command

Name

definenode - Defines the nodes in a cluster.

Synopsis

definenode [**−h**]

definenode [**−v**] **−f** *nodedef*

definenode [**−v**] [**–m** *install_method*] [**−n** *starting_node*] [**−c** *count*] [**−H** *HWControlPoints*] [**−C** *ConsoleServers*] [**−t** *HWtype*] [**−m** *install_method*]

Description

The **definenode** command runs on the management server and defines all the nodes in the cluster. Before running this command, you must run the **installms** command to install the cluster management server. This command can be rerun but cannot redefine nodes that are already defined. It does not actually install the nodes. The node installation is done by the **installnode** command.

Note that the **addnode** command runs the **definenode**, **setupks**, and **installnode** commands automatically.

If you do not provide some of the arguments, the program prompts you for each piece of information that it needs.

The **definenode** command generates the **/etc/opt/csm/netfinity_power.config** file. A template file is shipped in **csm.server**. See the **/etc/opt/csm/netfinity_power.config.tmpl** file This template file contains a default user ID and password, which is supplied for the service processor when it is shipped from the factory. The **definenode** command uses the default user ID and password from the template file to generate the user ID and password for all of the nodes in the cluster. To change the password or user ID for a node from the default, the file created from the template when **definenode** is run, **netfinity_power.config**, must be edited manually after **definenode** is run. Additional steps for changing these user IDs and passwords are supplied in *IBM Cluster Systems Management for Linux Remote Control Guide and Reference*.

Notes:

1. You cannot redefine a node that is already defined. If you attempt to define a node that is already defined by using the **−n** option, the **definenode** command fails and returns an error message stating that the node is already defined. The command can be rerun and the defined node can be avoided when the command is rerun. But, if a defined node is listed in a *nodedef* file, processing is completed for the nodes that are undefined, and an error message is generated for the already defined nodes.
2. The **definenode** command does not add the host names to the nameserver or to the **/etc/hosts** file on the management server. You must add the host name to the nameserver or to **/etc/hosts** manually before **definenode** is run.

The **definenode** command creates PreManagedNode objects based upon user input in the CSM database (in the PreManagedNode class) for each node. To see the PreManagedNode objects, type:

```
lsnode -P
```

See the **lsnode** man page for details about the attributes that are added to the CSM database.

You can list the attributes of a premanaged node by typing:

```
lsnode -APl nodename
```

You can change the attributes of a defined node by using the **chnode** command.

Options

−c *count*

Specifies how many nodes to define. IP addresses are calculated for nodes beyond the IP address for the starting node. The default is one.

−C *ConsoleServers*

Specifies the list of console server definitions. Multiple console server definitions are separated by commas. To specify that there are no console servers, enter a null value as follows:

```
-C " "
```

Note: A blank space is required between the opening and closing double quotation marks.

Each entry is formatted as follows:

```
-C csname[:method:csnum:starting_portnum][, ...]
```

where:

csname

Represents the host name or IP address of a console server.

method

Represents the console method. The default method is **esp**. Valid values for this field are `"esp"`, `"conserver"`, or `"none"`, or the field can be left blank. If the console server name field is left blank (the **-C** option has a value of ″ ″), then the value assumed for this field is `"none"`. If the console server name field is not blank and this field is blank, the default value of `"esp"` is assumed.

csnum Represents the console server number. The default is 1.

starting_portnum

Represents the starting console port number. The default is 0. If some ports have already been defined for the console server, then a starting port number higher than 0 needs to be used. The console port number is represented by a single hexadecimal character (0–f). The number of ports available to each console server depends on the console method.

If the console method is the default **esp**, the program determines how many ports are available. If you would like to know this information, type:

```
esptty -c
```

Otherwise, the default is 16. The port number is incremented up to the number of nodes to be defined. Each node is assigned a specific port number and console server. All the ports for a console server are assigned, and then the ports for the next console server are assigned. The console servers are used in the order that they are specified with the **-C** option.

If any of the port numbers for a console server are already in use by an existing ManagedNode or PreManagedNode, the **addnode** command fails without defining any nodes.

For more details on console servers, hardware control points, and service processors, see *IBM Cluster Systems Management for Linux Remote Control Guide and Reference*.

−f *nodedef*

Represents the node definition file. This file contains the list of node names, the console server definitions, and the hardware control point definitions. A sample node definition file is supplied in **/opt/csm/install/nodedef.sample**. This option cannot be used with any other option except **−v**. See the **nodedef** man page for more information.

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

−H *HWControlPoints*

Specifies the list of hardware control points and their associated service processors. Multiple hardware control point definitions are separated by commas. To specify that there are no hardware control points, enter a null value as follows:

```
-H " "
```

Note: A blank space is required between the opening and closing double quotation marks.

Each hardware control point definition for a node is specified in the following format:

```
-H HWCtrlPt[:method:service_processor_name][,... ]
```

where:

HWCtrlPt

Represents a hardware control point by host name or IP address.

method

Represents the power method. Valid values are `netfinity` or `none`, or the field can be left blank. If the hardware control point is listed and the hardware type is `netfinity`, then the default is `netfinity`. If the hardware control point field is not listed (the **-H** optionhas a value of *" "*) or the hardware type is not `netfinity`, then the default for this field is `none`.

service_processor_name

Represents the starting internal service processor (ISP) name. A maximum of 10 service processors are connected to each hardware control point. If a starting service processor name is not provided, the default is **node01**. There are two formats provided for the service processor name. The first format is **node***nn*, where *nn* is an integer from 01 through 10. A starting service processor higher than **node01**needs to be used if some of the service processors have already been defined for a hardware control point. The second format is based on the short host name of each node and is specified by entering the word **hostname** as the value for *service_processor_name*.

Note: This changes the setting of the CSM database **SvcProcName** attribute name of the ISP to the short host name of the node. To change the actual ISP name to the short host name of the node, refer to the *IBM Cluster Systems Management for Linux Remote Control Guide and Reference*. It is worthwhile to change the ISP to the short host name, for example, if you want to track SNMP (Simple Network Management Protocol) alerts. The ISP provides reporting mechanism support for tracking SNMP alerts.

The service processor name can be incremented up to the number of nodes to be defined. Each node is assigned to a specific service processor and hardware control point. All the service processors attached to a hardware control point are assigned, and then the service processors attached to the next hardware control point are assigned. The hardware control points are used in the order that they are specified with the **-H** option

If any of the hardware control points for a management service processor are already in use by an existing ManagedNode or PreManagedNode, then the **definenode** command fails without defining any nodes.

−m *install_method*

Specifies the installation method. There are two valid values, **csmonly** (to install just CSM) and **kickstart** (to install the operating system and CSM). The default is **csmonly**.

−n *starting_node*
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 ***count***. If a host name is specified, it is converted to an IP address.

−t *HWType*
Specifies the hardware type for all of the nodes to be defined. If the hardware type is set to `netfinity`, then PowerMethod default to `netfinity`, provided the hardware control point is set.

−v Writes the command's verbose messages to standard output.

Examples

1. To be prompted for all the necessary information, type:

   ```
   definenode
   ```

   The program requests the required information from you.
2. To use a node definition file to specify the nodes to be defined, type:

   ```
   definenode -f nodedef
   ```
3. To add one node for CSM only installation to an existing cluster, type:

   ```
   definenode -nclsn36
   ```
4. To add 16 nodes for CSM only installation to a new cluster, type:

   ```
   definenode -nclsn02 -c16 -Hmgt03,mgt04 -Cmgt01 -tnetfinity
   ```
5. To add four nodes for CSM only installation to an existing cluster, type:

   ```
   definenode -nclsn18 -c4 -Hmgt04::node07 -Cmgt02 -tnetfinity
   ```
6. To define five nodes for a full installation and their associated hardware control points and console servers, with a hardware type of `netfinity`, type:

   ```
   definenode -nclsn02 -c5 -Hmgtn03 -Cmgtn02 -tnetfinity -mkickstart
   ```

   ```
   definenode: Adding CSM Nodes:
   definenode: Adding Node clsn02.ppd.pok.ibm.com(9.114.133.197)
   definenode: Adding Node clsn03.ppd.pok.ibm.com(9.114.133.198)
   definenode: Adding Node clsn04.ppd.pok.ibm.com(9.114.133.199)
   definenode: Adding Node clsn05.ppd.pok.ibm.com(9.114.133.200)
   definenode: Adding Node clsn06.ppd.pok.ibm.com(9.114.133.201)
   ```

Files

/opt/csm/bin/definenode	Location of the **definenode** command.
/etc/opt/csm/netfinity_power.config.tmpl	Location of the template file that contains a default user ID and password, which is supplied for the service processor when it is shipped from the factory. The **definenode** command uses the default userid and password from the template file to create user IDs and passwords for all the nodes in the cluster. The **definenode** command generates the **netfinity_power.config** file to contain these passwords.
/etc/opt/csm/netfinity_power.config	Location of the service processor password file. To change the user ID and password for a node, this file must be edited manually after **definenode** is run.
/opt/csm/install/nodedef.sample	Location of the sample node definition file.

See Also

- The **addnode**, **chnode**, **createnode**, **installms**, **installnode**, **makenode**, **rpower** commands.
- The **netfinity_power.config** file, **netfinity_power.config.tmpl** template, **nodedef** file
- *IBM Cluster Systems Management for Linux Planning and Installation Guide* (am7LXstp.pdf), *IBM Cluster Systems Management for Linux Remote Control Guide and Reference* (am7LXrem.pdf)

Author

Sean Safron - cluster@us.ibm.com

dmsctrl Command

Name

dmsctrl - Displays or changes certain parameters that affect the distributed management server of CSM.

Synopsis

dmsctrl [**−h**] [**−v**] [**−t** *HostResponseTimeout*] [**−i** *FpingInterval*] [**−p** *PowerStatusInterval*]

Description

The **dmsctrl** command, which is run on the management server, allows you to manipulate the parameters that CSM uses in the distributed management server. Specifically, parameters can be set or queried that affect the way **fping** is periodically run and how often the power status of each node is queried. The **fping** tool is an open source tool that pings a set of nodes in parallel. The **ping** packets are sent to all the nodes at the same time, and then the tool waits for all the responses. CSM runs **fping** at a specified interval to determine the network status of all the nodes in the CSM cluster. The latest status of each node is cached and made available by means of the **lsnode -p** command.

The power status is queried by the **rpower** command. The results are cached and made available by means of the command:

```
lsnode -a PowerStatus
```

One or more of the parameters can be set using the optionsthat follow. If no option is specified, the value of all parameters is displayed.

Options

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

−v Writes the command’s verbose messages to standard output, including messages from the DBI layer.

−i *FpingInterval*
Specifies the length of time (in milliseconds) between **fping** invocations. All nodes are **ping**ed in one **fping** invocation. This value should be at least twice the *HostResponseTimeout*. The default is 5000ms or 5 seconds.

−p *PowerStatusInterval*
Specifies the length of time (in minutes) between queries of the power status. The default is 5 minutes.

−t *HostResponseTimeout*
Specifies the length of time (in milliseconds) that **fping** waits for a return packet before assuming that a node is not responding. The default is 500ms.

Examples

1. To change the ping interval to .25 second and change the timeout interval to 3 seconds, type:
   ```
   dmsctrl -i 250 -t 3000
   ```
2. To change the power query interval to 3 minutes, type:
   ```
   dmsctrl -p 3
   ```
3. To show the values currently set for the ping interval and for the timeout, type:
   ```
   dmsctrl
   ```

Files

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

See Also

The **fping**, **lsnode**, **rpower** commands.

Author

Bruce Potter - cluster@us.ibm.com

dsh Command

Name

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

Synopsis

dsh −q

dsh [**−h**] [**−a**] [**−B** | **−P**] [**−c**] [**−i**] [**−m**] [**−v**] [**−z**] [**−l** *login_name*] [**−n** *host[,host...]*] [**−N** *node_group[,node_group...]*] [**−w** {*host[,host...*] | – }] [**−o** *″remote_shell_options″*] [**−r** *remote_shell_path*] [**−f** *fanout_value* | **−s**] [*command*]

Description

The **dsh** command runs on the management server but can be run from any node as long as security is set up to allow the node to run commands on other nodes. The **dsh** command invokes commands on a set of Linux nodes concurrently. It issues a Linux 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 the nodes can be managed. **/bin/rsh** is the model for syntax and security. It is assumed that the remote login shell of the user of **dsh** is the bash shell.

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

The first way is called the **node list**. The node list is obtained from the first existence 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.

This is the preferred manner.

The second way is called the **working collective**. The working collective is obtained from the first existence 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.

If neither a node list nor a working collective exists when this approach is used, an error has occurred, 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, type `exit` or press Enter 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, **/usr/ucb:/bin:/usr/bin**. (For example, to set **DSH_PATH** to the path set on the source node, use **DSH_PATH=$PATH**).

The maximum number of concurrent remote shell commands can be specified with the fanout (**-f**) option or by means of 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 rather than 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 failed. This has nothing to do with the exit code of the remotely issued command. If a remote shell fails, 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.

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 fail, the **dsh** exit value is greater than 0. The exit value is increased by 1 for each remote shell failure.

No particular error recovery for command failure 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.

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 down (for example, a remote shell command issues a nonzero return code), subsequent commands are not sent to this node on this invocation of **dsh** 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, for example, a remote secure shell that conforms to the IETF (Internet Engineering Task Force) Secure Shell 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 execution 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.

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

Options

−a Adds all nodes defined to IBM Cluster Systems Management for Linux (CSM) to the node list.

−B Allows nodes specified by the **−a**, **−n**, and **−N** options to be in both the PreManagedNode table/class and the ManagedNode table/class. This option cannot be used with the **−P** option. If no options are specified, nodes are selected by default from the ManagedNode table/class only.

−c Specifies that commands that failed continue to be sent to the remote nodes for execution.

−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 execution 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 out.

−i Informs the user that a node is not responding and prompts the user as to 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.

−m Prints the results of monitoring for each node in the form of the starting and completion messages for each node.

−n {*host[,host...] | – }*
Specifies a list of host names, separated by commas, to include in the node list. If ″-″ is specified, you enter standard input mode. You know that you are in standard input mode because a new line is provided that has no **dsh** prompt. Enter the host names a 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.

Note: Duplicate host names are included only once in the node list.

−N *node_group[,node_group...]*
Resolves one or more CSM-specified 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.

−P Allows nodes to be specified from the PreManagedNode table/class only. This option cannot be used with the **−B** option. If no options are specified, nodes are selected by default from the ManagedNode table/class only.

−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 exist on the **dsh** command line alone. It cannot be used in conjunction with any other **dsh** option or with the *command* argument.

−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**.

−s Specifies output in streaming mode. The output is unsorted, but performance is likely to improve, and memory utilization is reduced.

−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 **lsnode** 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 10 seconds to check the node that is not responding, rather than the minute typically taken for the remote shell command to time out.

−w {*host[,host...] | – }*
Specifies a list of host names, separated by commas, to include in the working collective. If ″-″ is specified, you enter standard input mode. You know that you are in standard input mode because

a new line is provided that has no **dsh** prompt. Enter the host names a 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.

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

−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

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 remote shell 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_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.

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

Security

Security considerations are the same as for the remote shell command.

Examples

1. To issue 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 the **ps** command on the node list and filter results locally, enter:
```
dsh ps -ef | grep root
```
7. 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
```

8. 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
   ```

9. 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
   ```

10. 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
    ```

    The 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
    #
    ```

Files

/opt/csm/bin/dsh
Location of the **dsh** command

/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.

See Also

The **dshbak**, **rsh** commands.

Author

John Simpson - cluster@us.ibm.com

dshbak Command

Name

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

Synopsis

dshbak [**−c**]

Description

The **dshbak** command runs on the management server but can be run from any node as long as security is set up to allow the node to run commands on other nodes. 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.

Options

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

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
```

Diagnostics

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.

Files

/opt/csm/bin/dshbak	Location of the **dshbak** command

See Also

The **dsh** command.

Author

John Simpson - cluster@us.ibm.com

getmacs Command

Name

getmacs - Automatically gathers and stores MAC addresses in the CSM database.

Synopsis

getmacs [**−h**] [**−v**] *node_list*

Description

The **getmacs** command automatically gathers MAC addresses and stores them in the *Macaddr* attribute of the PreManagedNode object in the CSM database. It is called by **setupks** to gather MAC addresses during installation, but it can also be used manually by an administrator to update the database, for example, when a network adapter card is changed manually for a node.

The following prerequisites must be met before running the **getmacs** command:

1. Before running a full installation, set the node boot order to the following:
   - Floppy
   - CD-ROM
   - Network
   - Hard disk
2. The **definenode** command must be run with the *HwControlPoint* and *ConsoleServer* attributes provided before **getmacs** is run. This is because a node must have the service processor and remote console attributes defined in order for **getmacs** to gather its MAC address; that is, the **rpower** and the **rconsole** commands must already having been set up for all nodes.

   See the **rpower** and **rconsole** man pages and the *IBM Cluster Systems Management for Linux Remote Control Guide and Reference* for more details on the remote power commands.
3. The **setupks** command must be run before **getmacs** can be run manually.

The **getmacs** command reboots each node to gather its MAC address. If a *Macaddr* value is already defined for a node and **getmacs** is run, the original *Macaddr* value is overwritten with the new MAC address.

Note: When **getmacs** is called and run by **setupks,** it does not overwrite an existing *Macaddr* value.

The *node_list* parameter is a space-separated list of ManagedNode or PreManagedNode objects. If a node list is not specified, **getmacs** attempts to gather MAC addresses for all PreManagedNode objects. If a MAC address already exists, it will be overwritten.

Options

−h	Displays usage information to standard output.
−v	Writes the command’s verbose messages to standard output.

Files

/opt/csm/bin/getmacs	Location of the **getmacs** command
/tftpboot/pxelinux.cfg/HEX	Location of the pxelinux configuration file. HEX represents the node IP address as a hexadecimal value.

See Also

The **definenode**, **setupks** commands.

IBM Cluster Systems Management for Linux Planning and Installation Guide for more information on the installation processes.

IBM Cluster Systems Management for Linux Remote Control Guide and Reference for more information about the remote control commands.

Author

Vallard Benincosa - cluster@us.ibm.com

installms Command

Name

installms - Installs the CSM management server.

Synopsis

installms [**−h**]

installms [**−f**] **[−v**] [**−p** *pkg_path*]

Description

The **installms** command is copied to a directory on your system and run from there. It should not be run directly from the CSM distribution CD-ROM. The **installms** command is run on the management server and copies the required packages for installation from CD-ROM or a directory specified by the **−p** option into the appropriate **/tftpboot** directory and then installs the management server for a cluster of nodes. The packages to be copied are as follows:

- CSM and RSCT RPMs are copied from the CSM CD-ROM to **/tftpboot/rpm**.
- Tarball packages are copied from the CSM CD-ROM to **/tftpboot/tarball**.
- Other RPMs are copied from CD-ROM to **/tftpboot/rpm**.
- The Red Hat Linux RPMs are copied from the Red Hat CD-ROMs to **/tftpboot/rpm**. The Red Hat RPMs are used to install missing dependencies on the management server and the nodes.

The **installms** command assumes that an installation of the Linux distribution has been performed on your management server. The current default is Red Hat Linux 7.1. All the required RPMs for CSM as well as CSM itself are installed by **installms**. The RPMs are copied to **/tftpboot/rpm** and the tarballs are copied to **/tftpboot/tarball**. The **installms** command links the tarball files from **/tftpboot/tarball** to **/opt/csm/reqs** and then installs the tarball files by using the scripts shipped with **csm.core** and **csm.server**.

Options

−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 directories, separated by colons (*p1*:*p2*:*p3*), where RPMs and tarballs can be found. The default is **/mnt/cdrom**. CSM RPMs are found in the specified directories. Red Hat RPMs are found in the **/RedHat/RPMS** subdirectory of the specified directories. tarballs are found in the **/reqs** subdirectory of the specified directories. If Red Hat RPMs cannot be found in one of the specified directories, **installms** prompts you for the Red Hat CD-ROMs.

−v Writes the command’s verbose messages to standard output.

Examples

1. To install the management server of a cluster, type:

```
installms
```

2. To install the management server of a cluster by using a CSM distribution that is available in a directory instead of a CD-ROM, type:

```
installms -p /tmp/distrib
```

Files

/mnt/cdrom/installms	Location of the **installms** command when it is available on the distribution CD-ROM so that the CSM server does not need to be installed first.
/opt/csm/bin/installms	Location of the **installms** command.

See Also

- The **addnode**, **createnode**, **definenode**, **installnode**, **makenode**, **monitorinstall**, **setupks** commands.
- *IBM Cluster Systems Management for Linux Planning and Installation Guide* (am7LXstp.pdf)

Author

Sean Safron - cluster@us.ibm.com

installnode Command

Name

installnode - Installs the nodes in a cluster.

Synopsis

installnode [**−f**] [**−h**] [**−v**] [**−U**] [*node_list*]

Description

The **installnode** command runs on the management server and performs a CSM-only installation or a full installation on each PreManagedNode object, depending on the value of the *InstallMethod* attribute of the node (**csmonly** or **kickstart**). ManagedNode objects can be installed also if the **−U** option or a node list is specified. For a CSM-only installation, the **installnode** command runs synchronously; that is, when the command is completed, the cluster is ready to use. For a full installation, the **installnode** command runs asynchronously; that is, after the installation process is initiated by rebooting the node, **installnode** exits.

Note: The **addnode** command can be used to automatically run this command for a standard installation when the default values for the **installnode** arguments are sufficient.

Depending on the *InstallMethod* attribute, **installnode** performs a CSM-only installation or installs the whole operating system along with CSM on that node. The attribute is set by **definenode** or **addnode** and checked by **setupks** to determine the nodes that require full installation of the operating system and CSM. Both types of installations can be done during the same run. Full installations are initiated first, and then CSM-only installations are initiated. After all of the full installations have been initiated and the CSM-only installations are complete, the **installnode** command is complete.

The following logs provide a history of the installation process. The **/var/log/csm/makenode.log** is located on each node and records what happens to the node as it is installed. The **/var/log/csm/installnode.log** is located on the management server and provides detailed information on a CSM-only installation and a list of the nodes for which installation was attempted on a full installation. Run **dsh** to see the **makenode.log** file for detailed information on the activity that took place during installation as the first step in debugging any problems. For a full installation, see the **/var/log/csm/kickstart.log** file for information on the processing of the kickstart post-installation script, and the **/var/log/csm/firstboot.log** file for information on the processing of the post-installation first boot to its hard drive. The **monitorinstall** command also displays install status.

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

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

1. NFS must be available for mounting **/tftpboot** on the nodes.
2. For a CSM-only installation, the **dsh** command must be available to perform remote commands on the nodes; that is, security must be set up on each node in such a way that **dsh** is allowed to run commands on that node.

During a Kickstart install, **installnode** does the following:

1. Reboots the node.
2. As the node reboots, broadcasts its MACaddress.
3. The **dhcp** server on the management server accepts the node’s **dhcp** request and **pxelinux** is used to initiate a Kickstart install. **installnode** manages a **pxelinux** configuration file for each node in **/tftpboot/pxelinux.cfg/<node-ip-addr-in-hex>**.
4. After the Kickstart install of the OS is complete, the Kickstart post-install script (found in **/tftpboot/ks71/<node-ip-addr>-kickstart**) is run. After some additional configuration, the Kickstart

post-install script adds the **firstboot** script (**/tftpboot/bin/<node-hostname>.firstboot**) to **/etc/inittab**. It then modifies the **pxelinux** configuration file on the management server (**/tftboot/pxelinux.cfg/<node-ip-addr-in-hex>**) so the next reboot of the node reboots from the local hard drive.

5. The node is rebooted to the local hard drive.
6. As the reboot completes, the **firstboot** script (which was listed in **/etc/inittab**) is run.
7. The **firstboot** script runs **makenode** (which installs CSM and its dependencies and sets its management server), and then removes itself from **/etc/inittab**.
8. The node is now ready for use.

To install CSM only, **installnode** does the following:

1. Uses **dsh** to perform an **NFS** mount of **/tftpboot** to each node.
2. Uses **dsh** to run **makenode** from the mounted **/tftpboot** on each node.

To perform a CSM-only installation, you must run the **installms** and **definenode** commands to install and set up the cluster management server.

Before running a full installation, set the node boot order to the following:

- Floppy
- CD-ROM
- Network
- Hard disk

To perform a full installation, **installms**, **definenode** and **setupks** must be run before **installnode** is run in order to install the operating system and set up the cluster management server.

The *node_list* parameter is a space-separated list that contains either ManagedNode or PreManagedNode objects, or both.

Note: The following discussion shows what actions the **installnode** command takes, depending on whether the *node_list* parameter or the **−U** option is specified or whether the command is issued without parameters or options.

installnode
Does a CSM-only or full installation of all PreManagedNode objects depending on the *InstallMethod* attribute on each node.

installnode *node_list*
Does a CSM-only or full installation of the list of ManagedNode and PreManagedNode objects depending on the *InstallMethod* attribute on each node.

installnode −U
Does a CSM-only update of all ManagedNode objects even if the *InstallMethod* for a node is set to **kickstart**.

installnode −U *node_list*
Does a CSM-only update of the list of PreManagedNode and ManagedNode objects even if the *InstallMethod* for a node is set to **kickstart**

To see the progress of an installation, run the **monitorinstall** command.

Options

−f Forces installation regardless of the status of the existing installation.

−h Writes command’s usage statement to standard output.

−U	(See the note on command behavior in the *Description*.)
−v	Writes command’s verbose messages to standard output.

Environment

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

CSM_FANOUT_DELAY
Sets the delay in seconds between rebooting groups of nodes. If this is not set, the delay is 10 seconds.

Examples

1. To install all the nodes that are in the PreManagedNode class of a cluster, run this command from the cluster management server:
   ```
   installnode
   ```
2. The following command runs a CSM-only or full install of nodes clsn02 and clsn03 depending on the InstallMethod attribute for each node. Note that each node may be either a PreManagedNode or a ManagedNode:
   ```
   installnode clsn02 clsn03
   ```
3. The following command runs a CSM-only update of nodes clsn02 and clsn03 regardless of the InstallMethod attribute set for each node:
   ```
   installnode -U clsn02 clsn03
   ```

Files

/tftpboot/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. The oldest file is **installnode.log.4**.
/var/log/csm/kickstart.log	Location of the log file on each node when a full installation is performed and the Kickstart post-installation script is run.
/var/log/csm/firstboot.log	Location of the log file on each node when a first boot is performed.
/var/log/csm/makenode.log	Location of the log file on each managed node for the **makenode** command. Up to five copies of this log are maintained. Old logs receive a numeric suffix up to 4. The oldest file is **makenode.log.4**.

See Also

- The **addnode**, **createnode**, **definenode**, **installms**, **makenode**, **monitorinstall**, **setupks** commands.
- The **kscfg.tmpl** File
- *IBM Cluster Systems Management for Linux Planning and Installation Guide* (am7LXstp.pdf), *IBM Cluster Systems Management for Linux Remote Control Guide and Reference* (am7LXrem.pdf)

Author

Sean Safron - cluster@us.ibm.com

kscfg.tmpl File

Name

kscfg.tmpl - Linux operating system configuration file generated by **setupks**.

Description

This file is the template that is used by **setupks** to create a Kickstart configuration file. The template can be found at **/opt/csm/install/kscfg.tmpl**. The Kickstart configuration file that is generated by **setupks** from this template contains configuration information that is gathered by Kickstart during the installation of the Linux operating system. This file can be used as is, or it can be modified by the user. See an annotated sample template in the Appendix of *IBM Cluster Systems Management for Linux Planning and Installation Guide* for instructions on how to properly modify the template.

Note: For the current version of the template, see **/opt/csm/install/kscfg.tmpl** on your system.

To affect the settings of all the nodes in the cluster, modify the Kickstart configuration file template. Do this before running **setupks**.

To change node-specific settings, modify the generated Kickstart configuration file for a specific node. The generated node-specific configuration files are called **/tftpboot/ks71/***node_ip_address***-kickstart**. Modify the generated node-specific configuration file after running **setupks**.

Note that 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 variables that are automatically customized:

#MGMTSVR_HOSTNAME#
Replaced with the host name of the management server.

#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, **/tftpboot/rh71**.

The netmask, gateway, and nameserver values are represented by attributes that can be specified on the **setupks** 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, be careful not to alter sections that are surrounded by the following text:

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

Files

/opt/csm/install/kscfg.tmpl	Location of the Kickstart configuration file template.
/tftpboot/ks71/*node_ip_address***-kickstart**	Location of node-specific Kickstart configuration file

See Also

- The **addnode**, **definenode**, **rconsole**, **rpower**, **setupks** commands
- *IBM Cluster Systems Management for Linux Planning and Installation Guide* (am7LXstp.pdf)

- *IBM Cluster Systems Management for Linux Remote Control Guide and Reference* (am7LXrem.pdf)
- *The Official Red Hat Customization Guide* for the Kickstart file format

Author

Sean Safron - cluster@us.ibm.com

lsnode Command

Name

lsnode - Lists the managed node definitions in the IBM Cluster Systems Management for Linux (CSM) database.

Synopsis

lsnode [**−h**] [**−i | −s | −p | −a** *attr,attr...* **| −A**] [**−B**] [**−P**] [**−l | −d** *delim* **| −D** *delim*] [**−x**] [**−v**] {*host ...* **| −w** ″**where select string**″ **| −N** *nodegroup*}

Description

The **lsnode** command is used primarily during installation and for diagnostic purposes. It can be run on any node, including the management server, and lists the managed nodes that are defined to IBM Cluster Systems Management for Linux (CSM). When either the **−a** or **−A** flag is specified, **lsnode** displays the following information about one specific node or all nodes respectively:

ConfigChanged	Specifies changes to one or more persistent attributes.
ConsoleMethod	Determines the program to invoke for a specific type of console server. The attribute value is one of two options: **esp | conserver**
ConsolePortNum	Console port number. For ESP, this must be a single hexadecimal digit within the range 0 - f.
ConsoleServerName	Console server host name; for example, **mgtn02.pok.ibm.com**.
ConsoleServerNumber	Console server number; for example, some number, 1 - *N*. For ESP, this is the ESP number associated with the Equinox ESP system.
Hostname	Host name of the node.
HWControlPoint	Host name of the Advanced System Management (ASM) PCI adapter to which the internal service processor (ISP) of this node is connected; for example, **mgtn03** for x330 or x340 nodes, or the Remote Supervisor Adapter (RSA) for an x342 node.
HWModel	Hardware model of the node. Currently unused.
HWSerialNum	Hardware serial number of the node. Currently unused.
HWType	Hardware type of the node
InstallDisk	Installation disk. Currently unused.
InstallDiskType	Installation disk type. Currently unused.
InstallMethod	Installation method. This attribute value is **csmonly** or **kickstart**.
LParID	Logical partition ID (This is applicable to the Power PC platform only).
Macaddr	MAC address.
OSDistribution	Operating system distribution. Currently unused.
OSKernel	Operating system kernel level. Currently unused.

OSType	Operating system type. Currently unused.
OSVersion	Operating system version. Currently unused.
PowerMethod	Determines the program to invoke for a specific type of hardware power control; for example, netfinity (which corresponds to **/opt/csm/bin/netfinity_power**).
PowerStatus	Determines whether the power is on or off.
Status	Represents the current accessibility status of the node. **Accessibility** is defined as the ability to successfully ping the node. The valid states are UNREACHABLE (0), REACHABLE (1), and UNKNOWN (127).
SvcProcname	Internal service processor name; for example, **node01**.
UniversalId	Universal ID (This is the node identifier.)

If **-w** *selectstr* is specified, **lsnode** uses that string in the ″where″ part of an SQL select statement against the database of nodes and displays information about the nodes that are matched. If no options are specified, all the nodes known to IBM Cluster Systems Management for Linux are displayed. Most of the options specify the information that is displayed for each node.

The **lsnode** command can retrieve the node information from several different places. Use **whichdb** to control where **lsnode** obtains information.

Options

−a *attr*	Displays the specified attribute.
−A	Displays all attributes.
−B	Lists nodes from the ManagedNode table/class and the PreManagedNode table/class. This option cannot be used with the **−p** option.
−d *delim*	Specifies the delimiter used to separate items within rows and between rows.
−D *delim*	Specifies the delimiter that should be used to separate items within a row. The default is a comma (,).
−i	Displays IP addresses.
−l	Displays output in long format. The **−x** option has no effect when this option is specified.
−N *nodegroup*	Displays the nodes that are in the specified node group. The node group is evaluated by the **nodegrp** command.
−p	Displays the status of the nodes.
−P	Lists nodes from the PreManagedNode table/class. (The default lists nodes from the ManagedNode table/class.) This option cannot be used with the **−p** option.
−s	Displays the short host name.
−v	Writes the command’s verbose messages to standard output.
−w *selectstr*	Displays 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). As a convenience, ″*″ means all nodes, as if a ″where″ string were not specified. If **whichdb** is set to **rmc**, then the syntax for the select string is described in the section ″Using Expressions″ in the *IBM Cluster Systems Management for Linux Administration Guide*. If **whichdb** is set to **file**, then normal SQL syntax is used.

−x Specifies not to 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	An invalid combination of options and arguments has been entered.
12	Node not found.
13	Cannot show ping status.
14	Cannot specify a node group or more than one node because the file database does not support it.
15	Specified node group is empty.
31	An IP address that could not be resolved to a host name was entered.
32	A host name that could not be resolved to an IP address was entered.
51	Badly formed select string. This error is returned from the DBI layer.

When an error is detected in the DBI layer, the return code of the DBI layer is used as the exit status. When RMC is used as the DBI layer, the exit status of the RMC command is returned as the exit status. The DBI (DataBaseInterface) is a perl interface to databases that is similar in purpose to ODBC. DBI drivers (DBDs) are available for many kinds of databases.

Examples

1. To list the names of all nodes, type:
   ```
   lsnode
   ```
2. To list the **ping** status of all nodes, type:
   ```
   lsnode -p
   ```
3. To list all of the attributes for node **websvr**, type:
   ```
   lsnode -Al websvr
   ```
4. To list all nodes whose host name is in the range between **clsn01** and **clsn09**, type;
   ```
   lsnode -w "Hostname like 'clsn0%'"
   ```

Files

/opt/csm/bin/lsnode	Location of the **lsnode** command

See Also

The **chnode**, **createnode**, **nodegrp**, **rmnode**, **whichdb** commands.

Author

Bruce Potter - cluster@us.ibm.com

lssensor Command

Name

lssensor - Displays the event sensor commands that have been added to RMC.

Synopsis

lssensor [**−a | h**] [**−v | V**] [**−n** *nodelist*] [*Name*]

Description

The **lssensor** command displays the attributes of the sensor identified by *Name*. If *Name* is omitted, the **lssensor** command lists the names of all the sensors defined to RMC. This command returns the error messages and exit status of the underlying **lsrsrc** RMC command.

The **lssensor** command can be run on the management server or on any node. It operates on the local node where it is run unless the CT_CONTACT environment variable is set.

See the **mksensor** command for a more complete explanation of RMC sensors.

Options

−a Displays all attributes of all sensors.

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

−n *nodelist*
When in high-availability (HA) cluster mode, this option specifies the node numbers for the nodes for which the sensors are listed. By default, sensors on all nodes are listed. To list the sensors defined on the local node only, set the CT_LOCAL_SCOPE environment variable to one before running this command. This option is not necessary when not in HA cluster mode.

−v Writes the command's verbose messages to standard output. This option can be entered either uppercase or lowercase.

−V Writes the command's verbose messages to standard output. This option can be entered either uppercase or lowercase.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0 Command has run successfully.

1 Error occurred with RMC.

2 Error occurred with CLI script.

3 Incorrect flag on command line.

4 Incorrect parameter on command line.

5 Error occurred with RMC that was based on faulty command line input.

Security

The user needs read permission for the IBM.Sensor resource class in order to run **lssensor**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. Lists the names of all of the sensors.
```
lssensor
```
2. Lists the names and attributes of all sensors.
```
lssensor -a
```
3. Lists the attributes of the sensor called NumLogins:
```
lssensor NumLogins
```

Files

/opt/csm/bin/lssensor	Location of the **lssensor** command

See Also

The **chsensor**, **lsrsrc**, **mksensor**, **rmsensor** commands.

IBM Cluster Systems Management for Linux Administration Guide for information on the ACL authorization file.

Author

Bruce Potter - cluster@us.ibm.com

makenode Command

Name

makenode - Runs on each CSM node to install CSM and its dependencies and to convert the node to a managed node.

Synopsis

makenode [**−h**]

makenode [**−f**] [**−v**] [**−M** *MapFile*] [**−n** *nodename*] **−m** *management_server*

Description

The **makenode** command runs on each node to install CSM and its dependencies. After installing CSM, **makenode** runs the **mgmtsvr** command to establish the relationship between the node and the management server and to convert the node from a premanaged node to a managed node.

The **makenode** command is normally run as part of the automated installation and is not used directly by the system administrator. For a CSM-only installation, it is called by **installnode** by means of **dsh**. For a full installation, **makenode** is called by the **firstboot** script. Note that Kickstart is only used to install the base Linux operating system. Then, **makenode** is used to install CSM and add the node to the cluster.

The **makenode** command can be rerun if necessary. Anything that has already been done is skipped. If an RPM is back-level or nonexistent, it will be upgraded or installed. Output is saved to the **makenode.log** on the node. For a full installation, output from the processing of the **firstboot** script is saved in **firstboot.log**.

The option **−n** *node_hostname* can be passed to **makenode** to ensure that the value which is defined in the *Hostname* attribute of the managed node is passed to **mgmtsvr**. This is needed if the output of **/bin/hostname** on the node does not resolve to the fully qualified host name or if the management server is not contacting the node through the primary network interface; for example, when the output of **/bin/hostname** returns the short host name but the managed node is defined with the fully qualified host name.

After installation is complete, remote RMC commands are enabled, and the managed node is ready to be managed actively. CSM commands can be run remotely from the node. For example, if **lsnode** is issued, all of the nodes in the cluster are displayed.

Options

−f Forces installation regardless of the status of the existing installation.

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

−m *management_server*
Specifies the management server, by host name or IP address, to which this managed node will belong.

−M *mapfile*
File that maps local node names to the host names by which the CSM management server knows the nodes. Normally, you do not need to specify this option because **installnode** creates a map file for all the nodes in the cluster.

−n *nodename*
Host name by which the CSM management server knows this node. Normally, you do not need to specify this option because it is used internally by **installnode** or **makenode** to ensure correct operation even when the host name value on the node resolves to a different name from that by which the management server knows the node. Defaults to the output of **/bin/hostname**.

−v	Writes the command’s verbose messages to standard output.

Examples

This command is not normally run by the user. It might be useful in certain diagnostic situations.

Files

/tftpboot/bin/makenode	Temporary location of the **makenode** command during installation.
/opt/csm/bin/makenode	Location of the **makenode** command.

See Also

- The **addnode**, **createnode**, **definenode**, **installms**, **installnode**, **monitorinstall**, **setupks** commands.
- *IBM Cluster Systems Management for Linux Planning and Installation Guide* (am7LXstp.pdf)

Author

Sean Safron - cluster@us.ibm.com

mksensor Command

Name

mksensor - Adds an event sensor command to RMC.

Synopsis

mksensor [**−h**] [**− v | V**] [**−i** *seconds*] [**−n** *nodelist*] *Name Command*

Description

The **mksensor** command adds the specified command to the Resource Monitoring and Control (RMC) subsystem to be run periodically to retrieve a user-defined value from the system. Usually this value is used to cause an event to occur by means of the Event Response resource manager (ERRM). For example, if the specified command returns the number of users logged on to the system (a variable none of the standard RMC resource managers return currently), an ERRM condition and response can be written to run a response when the number of users logged on exceeds a certain value. This enables administrators to extend the RMC monitoring capabilities without having to write a resource manager.

The specified command returns the value it retrieves from the system by sending it to standard output in the form Attr=value. The attribute name used depends on the type of the value and is one of: String, Int32, Uint32, Int64, Uint64, Float32, Float64, or Quantum. If only the value is sent to standard output, the attribute name is assumed to be String.

If more than one type of data is to be returned, a series of Attr=value pairs are sent to standard output, separated by blanks. For example: Int32=10 String=″abcdefg″. These attribute names are also the dynamic attributes that can be used in the ERRM condition. The exit value of the script is stored in the dynamic attribute ExitValue and can also be used in an ERRM condition. When the ERRM condition is written, the IBM.Sensor resource class should be specified, and the select string should normally be Name='*Name*', where *Name* is the name of the sensor being defined by this **mksensor** command.

The *Command* parameter represents the command that should be run periodically to retrieve the desired system value. By default, this command is run by RMC under the same user name that originally ran the **mksensor** command to add this sensor.

The **mksensor** command can be run on the management server or on any node. It normally operates on the local node where it is run unless the CT_CONTACT environment variable is set.

This command returns the error messages and exit status of the underlying **mkrsrc** RMC command.

Options

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

−i *seconds*
Specified how often to periodically run the command. This interval is in seconds. The minimum value is 10 seconds. The default value is 60 seconds.

−n *nodelist*
When in high-availability (HA) cluster mode, this option specifies the node numbers for the nodes on which this sensor is to be defined. To define the sensor on the local node only, set the CT_LOCAL_SCOPE environment variable to one before running this command. This option is not necessary when not in HA cluster mode.

−v Writes the command's verbose messages to standard output. This option can be entered either uppercase or lowercase.

−V Writes the command's verbose messages to standard output. This option can be entered either uppercase or lowercase.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred with RMC that was based on faulty command line input.

Security

The user needs write permission for the IBM.Sensor resource class in order to run **mksensor**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. Create a new sensor that runs a **numlogins** script to find out how many users are currently logged in:

   ```
   mksensor NumLogins /usr/local/bin/numlogins
   ```

   The **numlogins** script is similar to the following:

   ```
   #!/usr/bin/perl
   my @output='who';
   print 'Int32='scalar(@output), "\n";
   exit;
   ```

Files

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

See Also

The **chsensor**, **lscondition**, **lssensor**, **mkcondition**, **mkcondresp**, **mkresponse**, **mkrsrc**, **rmsensor** commands.

The **rmccli General Information** file for information on the syntax of attr=value syntax.

IBM Cluster Systems Management for Linux Administration Guide for information on the ACL authorization file and on ERRM and ERRM environment variables.

Author

Bruce Potter - cluster@us.ibm.com

mgmtsvr Command

Name

mgmtsvr − Displays or sets the management server entry for this managed node.

Synopsis

mgmtsvr [**−h**] [**−v**] [**−n** *nodename*] [**−d | −N |** *hostname* **|** *ipaddr*]

Description

The **mgmtsvr** command is run on the managed node and controls the management server that will manage this node in the CSM cluster. The node can be specified either by host name or IP address. When *host name* or *ipaddr* is specified, the **mgmtsvr** command sets the node represented by *host name ipaddr* or as the management server for this node and then notifies the newly specified management server that this node is now part of its cluster. The management server is also given the necessary authorization to this node.

After the management server is successfully set, you can run **lsnode** to display all the nodes in the cluster of this management server, and you can run **nodegrp** to display all of the node groups in the cluster of this management server, provided the management server is using RMC as the CSM database.

Note: In order for this command to be able to contact the management server, this node must already be in the RMC ACL list of the management server or have the appropriate key.

If *hostname* or *ipaddr* is not specified, the current management server is displayed.

Options

−d Deletes the management server entry.

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

−n *nodename*
Host name by which the CSM management server knows this node. Normally you do not need to specify this option because it is used internally by **installnode** or **makenode** to ensure correct operation even when the host name value on the node resolves to a different name from that by which the management server knows the node. Defaults to the output of **/bin/hostname**.

−N Displays the value for the local node name along with that of the management server. This argument is only valid when a host name or IP address for the management server is not specified.

−v Writes the command’s verbose messages, including messages from the DBI layer, to standard output.

Exit Status

0 Command has run successfully.

1 Error occurred when command was run.

Examples

1. To query the current CSM management server for this node, type:
```
mgmtsvr
```
2. To set the management server for this node to **csmsvr.com**
```
mgmtsvr csmsvr.com
```

Files

/opt/csm/bin/mgmtsvr	Location of **mgmtsvr** command
/var/log/csm/makenode.log	Location of the log output by the **makenode** command.

See Also

The **lsnode**, **nodegrp** commands.

IBM Cluster Systems Management for Linux Administration Guide for information on editing the RMC ACL file.

IBM Cluster Systems Management for Linux Planning and Installation Guide for FAQs on troubleshooting the RMC ACL file.

Author

Bruce Potter - cluster@us.ibm.com

---

monitorinstall Command

Name

monitorinstall - Displays the status of the CSM installation on each of the nodes.

Synopsis

monitorinstall [**−h**] [**−v**] [**−e**] [**−i**] [**−l**] [**−u**]

Description

The **monitorinstall** command displays the status of the CSM installation on each of the nodes. A filtering scheme can be used to display only nodes of interest. By default, all of the PreManagedNodes and ManagedNodes that are defined in the cluster are displayed in the installation monitor. You can choose to display all nodes, only the installed nodes, only the not-installed nodes, or only the nodes that failed to install. By default, if an option is not specified, all nodes are displayed. To run the **monitorinstall** command at specified intervals, use the **watch** command.

Options

−e Shows only the nodes that had an error during the installation process. Can be used with the **−i** or **−u** option to display different types of nodes. Note that some errors are undetectable by **monitorinstall**. If none of the options are specified, then all of the nodes are displayed.

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

−i Shows only the nodes that are installed. Can be used with the **−e** or **−u** option to display different types of nodes. Note that some errors are undetectable by **monitorinstall**. If none of the options are specified, then all of the 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. Can be used with the **−e** or **−i** option to display different types of nodes. Note that some errors are undetectable by **monitorinstall**. If none of the options are specified, then all of the nodes are displayed.

−v Writes the command’s verbose messages to standard output.

Examples

1. To display the progress of the installation, type:

   ```
   monitorinstall
   ```

   Output similar to the following is displayed:

   ```
   Node                         Status
   --------------------------------------------------------
   clsn02.ppd.pok.ibm.com       Installed
   clsn03.ppd.pok.ibm.com       Rebooting to Install Node
   clsn04.ppd.pok.ibm.com       Starting Kickstart Post-Install
   clsn05.ppd.pok.ibm.com       Installed
   clsn06.ppd.pok.ibm.com       Installed
   clsn07.ppd.pok.ibm.com       Starting Firstboot
   clsn08.ppd.pok.ibm.com       Not Installed
   ```
2. To display expanded details on the progress of the installation, type:

   ```
   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
```

Files

/opt/csm/bin/monitorinstall	Location of **monitorinstall** command.
/tftpboot/status/<node_hostname>	Location of the node’s install status file.

See Also

The **addnode**, **installnode**, and **watch** commands.

IBM Cluster Systems Management for Linux Planning and Installation Guide for more information on the installation processes.

Author

Vallard Benincosa - cluster@us.ibm.com

netfinity_power.config File

Name

netfinity_power.config File – The configuration file containing the passwords for the Netfinity internal service processors (ISPs).

Description

The **netfinity_power.config** file is an input file for the **rpower** command. The file is used during two login phases; the login to the Advanced System Management (ASM) PCI adapter, and the subsequent login to the target internal service processors. For security purposes, this file should be defined as read and write by root only (`6 0 0`). Review this file to verify that all nodes are defined properly. This file must be updated when a password has changed.

The **definenode** command creates the **netfinity_power.config** file automatically with default user IDs and passwords. If you are not using the defaults, you should change these user IDs and passwords in the **netfinity_power.config** file after running **definenode** for the first time. When **definenode** is run again, it checks to see if entries exist for the *HWControlPoint* and *SvcProcName* attributes. If none exist, the command creates entries for these attributes.

The **−H** option on **definenode** changes the setting of the CSM database *SvcProcName* attribute name of the ISP to the short host name of the node. To change the actual ISP name to the short host name of the node, refer to the *IBM Cluster Systems Management for Linux Remote Control Guide and Reference*. It is worthwhile to change the ISP to the short host name, for example, if you want to track SNMP (Simple Network Management Protocol) alerts. The ISP provides reporting mechanism support for tracking SNMP alerts.

See the **definenode** man page for more information on defining service processor names. See the **rpower** man page for more information on defining passwords.

The **netfinity_power.config** file is a flat text file, with each line containing four colon-separated attributes that map to attributes defined in CSM:

1. Hardware Control Point (`HWControlPoint`)
2. Internal Service Processor name (`SvcProcName`)
3. User ID (`id`)
4. Password (`passwd`)

These four attributes appear in the **netfinity_power.config** file in the following format:

```
HWControlPoint:SvcProcName:id:passwd
```

The first line of the file describes the ASM adapter for that group of 10 nodes. So, in the first line the `HWControlPoint` and `SvcProcName` attribute values will always be the same. You must have an entry for every node and ASM PCI adapter. See the next section for an example.

Examples

The following **/etc/opt/csm/netfinity_power.config** file example shows the attributes for two groups containing an ASM PCI adapter and 10 nodes in each group. The first line describes the ASM PCI adapter for that group and subsequent lines describe each node. The second column can be either an ASM PCI adapter or an ISP adapter.

```
mgtn03.ppd.pok.ibm.com:mgtn03.ppd.pok.ibm.com:USERID:PASSW0RD
mgtn03.ppd.pok.ibm.com:node01:USERID:PASSW0RD
mgtn03.ppd.pok.ibm.com:node02:USERID:PASSW0RD
mgtn03.ppd.pok.ibm.com:node03:USERID:PASSW0RD
mgtn03.ppd.pok.ibm.com:node04:USERID:PASSW0RD
mgtn03.ppd.pok.ibm.com:node05:USERID:PASSW0RD
```

```
mgtn03.ppd.pok.ibm.com:node06:USERID:PASSW0RD
mgtn03.ppd.pok.ibm.com:node07:USERID:PASSW0RD
mgtn03.ppd.pok.ibm.com:node08:USERID:PASSW0RD
mgtn03.ppd.pok.ibm.com:node09:USERID:PASSW0RD
mgtn03.ppd.pok.ibm.com:node10:USERID:PASSW0RD
mgtn04.ppd.pok.ibm.com:mgtn04.ppd.pok.ibm.com:USERID:PASSW0RD
mgtn04.ppd.pok.ibm.com:node01:USERID:PASSW0RD
mgtn04.ppd.pok.ibm.com:node02:USERID:PASSW0RD
mgtn04.ppd.pok.ibm.com:node03:USERID:PASSW0RD
mgtn04.ppd.pok.ibm.com:node04:USERID:PASSW0RD
mgtn04.ppd.pok.ibm.com:node05:USERID:PASSW0RD
mgtn04.ppd.pok.ibm.com:node06:USERID:PASSW0RD
mgtn04.ppd.pok.ibm.com:node07:USERID:PASSW0RD
mgtn04.ppd.pok.ibm.com:node08:USERID:PASSW0RD
mgtn04.ppd.pok.ibm.com:node09:USERID:PASSW0RD
mgtn04.ppd.pok.ibm.com:node10:USERID:PASSW0RD
```

Files

/etc/opt/csm/netfinity_power.config
Location of the **netfinity_power.config** file.

See Also

The **definenode**, **rpower** commands.

The *IBM Cluster Systems Management for Linux Remote Control Guide and Reference* (am7LXrem.pdf)

Author

John Simpson – cluster@us.ibm.com

nodedef File

Name

nodedef - CSM node definition file.

Description

The **nodedef** file is a text file that contains a list of attributes about a CSM node. This file is used as input to the **addnode** and **definenode** commands.

See *IBM Cluster Systems Management for Linux Remote Control Guide and Reference* for more details about the fields that comprise the entries in this file. There is one entry per line, and each line has the format:

```
name:HWControlPoint:PowerMethod:SvcProcName:ConsoleServerName:ConsoleMethod:ConsoleServerNumber: /
ConsolePortNum:HWType:InstallMethod
```

The fields that comprise the entries in this file represent the following:

name
Name of the node. This can be either a host name or an IP address and is assigned to the **Hostname** attribute in the CSM database. If necessary, it is converted to a host name.

HWControlPoint
Hardware control point. This can be either a host name or an IP address. If this field is left blank, there is no hardware control point.

PowerMethod
Determines the program to invoke for a specific type of hardware power control. The valid entries are `netfinity`, `none`, or the field may be left blank. If the field is left blank, then the default depends on the value of the hardware control point field. When the hardware control point field is blank, the default method is `none`. When the hardware control point field has a value and the HWtype is `netfinity`, the default method is `netfinity`. If the hardware control point field is blank or the HWtype is not `netfinity`, then the default method is `none`.

SvcProcName
Service processor name. The format can be one of the following:

- node*nn*, where *nn* can have a value of 01 through 10
- **hostname**, which uses the node’s short host name
- a value that matches the real internal service processor (ISP) name (text id)

ConsoleServerName
Console server name, which can be a host name or an IP address. If left blank, there is no console server.

Console Method
Determines the program to invoke for a specific type of console server. Valid entries are **esp**, **conserver**, or **none**. The format is any string. The default is **esp** if the console server has a value, or the default is **none** if the console server does not have a value.

ConsoleServerNumber
Console server number. You need to enter the appropriate value. See *IBM Cluster Systems Management for Linux Remote Control Guide and Reference* for details.

ConsolePortNum
Console port number, which can be a hexadecimal value from 0 through f.

HWType
Hardware type. The format is any string. There is no default. If set to `netfinity`, PowerMethod also becomes `netfinity`, provided the hardware control point is set.

InstallMethod

Type of installation to perform, full (kickstart) or CSM only. Valid values are **csmonly** or **kickstart**. If left blank, the default is **csmonly**.

Examples

1. A sample node definition file is available in: **/usr/local/csm/install/nodedef.sample**.
2. A sample filled-in node-attribute table is available in *IBM Cluster Systems Management for Linux Remote Control Guide and Reference*.

Files

/usr/local/csm/install/nodedef.sample

Location of a sample node definition file.

See Also

- The **addnode**, **definenode**, **rconsole**, **rpower** commands
- *IBM Cluster Systems Management for Linux Remote Control Guide and Reference* (am7LXrem.pdf), *IBM Cluster Systems Management for Linux Planning and Installation Guide* (am7LXstp.pdf)

Author

Sean Safron - cluster@us.ibm.com

nodegrp Command

Name

nodegrp - Manages node group definitions in the IBM Cluster Systems Management for Linux (CSM) database.

Synopsis

nodegrp {**−h −p | −a** *nodelist* **| −x** *nodelist* **| −D | −l | −w** *selectstr* **| −W −s** *node*} [**−d** *delim*] [**−P**] [**−B**] [*group*]

nodegrp {**−l | −s** *node* **| −h** } [**−v**] [**−d** *delim*]

Description

The **nodegrp** command can be run on any node, including the management server. It lists and updates the node groups that are defined to IBM Cluster Systems Management for Linux (CSM). 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 that string to apply against the node database each time the **nodegrp** command is invoked to display the members of the group.

Options can also be used to list all node groups, delete a node group, or remove nodes from a node group. The group specified as the input to the command is the group that is to be acted on for all options specified.

The *host* parameter can be specified by either a host name or an IP address.

The *group* parameter is the name of a specified node group.

Options

−a *nodelist*
Adds host names to a group, creating a group if necessary. A node list consists of one or more comma-separated host names.

−B Lists nodes from the ManagedNode table/class and the PreManagedNode table/class. This only has an effect if a select string is set with **−w**.

−d *delim*
Specifies the delimiter used to separate the items returned.

−D Deletes a group.

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

−l Lists all defined node groups.

−p Prints a group. If no option is specified, this is the default.

−P Lists nodes from the PreManagedNode table/class. This only has an effect if a select string is set with **-w**.

−s *host*
Displays all the group names (static and dynamic) that contain the node represented by this host name or IP address.

−v Writes the command’s verbose messages to standard output.

−w *selectstr*
Specifies the ″where″ clause of a select string to be used to search the nodes table in the

database to dynamically determine the list of nodes in the group. The group is created if it does not already exist. If **whichdb** is set to **rmc**, then the syntax for the select string is described in the section ″Using Expressions″ in the *IBM Cluster Systems Management for Linux Administration Guide*. If **whichdb** is set to **file**, then normal SQL syntax is used.

−W Displays the ″where″ clause of the select string set for this node group.

−x *nodelist*
Removes nodes from a group.

Exit Status

1	An invalid combination of options and arguments has been entered.
12	Group not found.
14	″where″ string set in attempt to add nodes to group.
15	″where″ string set in attempt to remove nodes from group.
16	Member list set in attempt to set the ″where″ string.
17	″where″ string not set in attempt to display the ″where″ string.
51	Badly formed select string. This error is returned from the DBI layer.

When an error is detected in the DBI layer, the return code of the DBI layer is used as the exit status. When RMC is used as the DBI layer, the exit status of the RMC command is returned as the exit status. The DBI (DataBaseInterface) is a perl interface to databases that is similar in purpose to ODBC. DBI drivers (DBDs) are available for many kinds of databases

Examples

1. To list all of the node groups, type:
   ```
   nodegrp
   ```
2. To create a node group called `g1` that contains `websvr1` and `websvr2`, type:
   ```
    nodegrp -a websvr1,websvr2 g1
   ```
3. To list the members of `g1`, type:
   ```
   nodegrp g1
   ```
4. To add a node (`websvr3`) to `g1`, type:
   ```
   nodegrp -a websvr3 g1
   ```
5. To remove `websvr2` from `g1`, type:
   ```
   nodegrp -x websvr2 g1
   ```
6. To create a node group called `ws` that contains all the nodes that start with `websvr`, type:
   ```
   nodegrp -w "Hostname like 'websvr%'" ws
   ```

Files

/opt/csm/bin/nodegrp	Location of the **nodegrp** command

See Also

The **chnode**, **createnode**, **lsnode**, **rmnode**, **whichdb** commands.

Author

Bruce Potter - cluster@us.ibm.com

predefined-condresp Command

Name

predefined-condresp − Defines default monitoring conditions and responses.

Synopsis

predefined-condresp [**−h**] [**−r**] [**−c | d**] [**−q**] [**−n**] [**−v**]

Description

The **predefined-condresp** command defines some monitoring conditions and responses that are useful to many administrators. These conditions and responses are used by the Event Response resource manager (ERRM) to monitor the cluster for conditions of interest. When the conditions occur, ERRM runs the corresponding responses. The **predefined-condresp** command is run automatically when **csm.server** is installed. This command can also be run again to restore the conditions and responses to their initial definitions. If run with no arguments, **predefined-condresp** removes the conditions and responses first and then defines them again. The options can be used only to define or only to remove the conditions and responses.

Options

−c	Defines conditions and responses by means of the actual ERRM commands that an administrator would use. This option is used in place of the **−d** option. Although performance is slower with this option, it shows by example how an administrator can define similar resources.
−d	Defines conditions and responses.
−h	Writes command's usage statement to standard output.
−n	Do not run the commands.
−q	Do not print the command before running it.
−r	Removes the conditions and responses.
−v	Writes the command's verbose messages to standard output.

Exit Status

0	Command has run successfully.
1	Error occurred when command was run.
n	The highest exit code that was returned by any of the ERRM commands.

Examples

1. To remove the current predefined conditions and responses and create them again, type:

```
predefined-condresp
```

2. To define the predefined conditions and responses, type:

```
predefined-condresp -d
```

Files

/opt/csm/bin/predefined-condresp
Location of **predefined-condresp** command

See Also

The **mkcondition**, **mkcondresp**, and **mkresponse** commands.

IBM Cluster Systems Management for Linux Administration Guide for information on ERRM and on the predefined conditions and responses that are available for monitoring.

Author

Bruce Potter - cluster@us.ibm.com

rconsole Command

Name

rconsole – provides remote console support for Cluster Systems Management (CSM) nodes.

Synopsis

rconsole [**−a**] [[**−B** | **−P**] [**−h**] [**−t**] [**−n** *host[,host...]*] [**−N** *Node_group[,Node_group...]*]

Description

The **rconsole** command provides remote console support for the nodes in a cluster. The command uses the CSM database to determine the nodes and their service processor information. The **−a** option causes the **rconsole** command to display a console for each node defined in the cluster. The **−n** and **−N** options cause the command to display a console for each node defined by these options. You can also define an environment variable RCONSOLE_LIST, which is the name of a file that contains a list of nodes to manage. The host names used must be the names specified when the nodes are defined in the CSM database.

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	fixed
2 to 4	5x8
greater than 4	nil2

Note: The nil2 font that is the default font when more that five systems are specified is not intended to be readable. Rather, it is intended to give a general idea of whether the node is up. Use RCONSOLE_FONT to override this default if you intend to read the information provided on these consoles.

Options

−a Runs the command on all of the nodes in the cluster.

−B Allows nodes specified by the **−a**, **−n**, and **−N** options to be in both the PreManagedNode table/class and the ManagedNode table/class. This option cannot be used with the **−P** option. If no options are specified, nodes are selected by default from the ManagedNode table/class only.

−h Writes usage information to standard output.

−n *Host[,Host...]*
Specifies a list of nodes on which to run the command.

−N *Node_group[,Node_group...]*
Specifies one or more node groups on which to run the command.

−P Allows nodes to be specifed from the PreManagedNode table/class only. This option cannot be used with the **−B** option. If no options are specified, nodes are selected by default from the ManagedNode table/class only.

−t Specifies text mode. Shows the contents of the remote console in the current **xterm** session. This option is allowed only when one node is specified.

Environment

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, one host name per line.

Exit Status

0 Command has run successfully.

1 Error occurred with one or more of the remote console commands.

Examples

1. To open a remote console to one system, type:
   ```
   rconsole -n clsn02
   ```
2. To open a remote console to a group of nodes defined in the CSM database as the node group **clients**, type:
   ```
   rconsole -N clients
   ```
3. To open consoles for a group of nodes with a specific font, type:
   ```
   export RCONSOLE_FONT=fixed10; rconsole -n clsn02,clsn03
   ```

Files

/opt/bin/rconsole	Location of the **rconsole** command.

See Also

The **espcfg**, **espdiag**, **esptty**, and **lsnode** commands.

IBM Cluster Systems Management for Linux Remote Control Guide and Reference

Author

John Simpson – cluster@us.ibm.com

rmnode Command

Name

rmnode - Removes a node definition from the IBM Cluster Systems Management for Linux (CSM) database.

Synopsis

rmnode [**−h**] [**−v**] [**−P**] *host*

Description

The **rmnode** command is run on the management server and deletes a managed node definition from the IBM Cluster Systems Management for Linux database. The managed node is represented by the *host* parameter, which can be specified by either host name or IP address. Removing a node from a cluster does not uninstall CSM and its prerequisites from the node. Rather, it disassociates the node from its management server. It removes the node from the database of the management server, and it informs the node that it is no longer attached to the management server.

Options

−h	Writes the command’s usage statement to standard output.
−P	Removes a node from the PreManagedNode table/class. (Normally, nodes are removed from the ManagedNode table/class.)
−v	Writes the command’s verbose messages to standard output.

Exit Status

1	An invalid combination of options and arguments has been entered.
12	Node not found.
51	Badly formed select string. This error is returned from the DBI layer. When an error is detected in the DBI layer, the return code of the DBI layer is used as the exit status. When RMC is used as the DBI layer, the exit status of the RMC command is returned as the exit status.The DBI (DataBaseInterface) is a perl interface to databases that is similar in purpose to ODBC. DBI drivers (DBDs) are available for many kinds of databases.

Examples

1. To remove the node named **websvr** from the CSM database, type:

   ```
   rmnode websvr
   ```

2. To remove the node with the IP address 9.117.10.51 from the database, type:

   ```
   rmnode 9.117.10.51
   ```

Files

/opt/cms/bin/rmnode	Location of **rmnode** command

See Also

The **chnode, createnode, lsnode, nodegrp, whichdb** commands.

The **rmccli General Information** file.

Author

Bruce Potter - cluster@us.ibm.com

rmsensor Command

Name

rmsensor - Removes an event sensor command from RMC.

Synopsis

rmsensor [**−h**] [**−v | V**] [**−n** *nodelist*] *Name*

Description

The **rmsensor** command removes the sensor identified by *Name* from RMC. This command returns the exit status and error messages of the underlying **rmrsrc** RMC command.

The **rmsensor** command can be run on the management server or on any node and normally operates on the local node where the command is run unless the CT_CONTACT environment variable is set to prevent this.

See the **mksensor** command for a more complete explanation of RMC sensors.

Options

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

−n *nodelist*
When in high-availability (HA) cluster mode, this option specifies the node numbers for the nodes from which this sensor is to be removed. To remove the sensor on the local node only, set the CT_LOCAL_SCOPE environment variable to one before running this command. This option is not necessary when not in HA cluster mode.

−v Writes the command's verbose messages to standard output. This option can be entered either uppercase or lowercase.

−V Writes the command's verbose messages to standard output. This option can be entered either uppercase or lowercase.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Security

The user needs write permission for the IBM.Sensor resource class in order to run **rmsensor**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.

3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred with RMC that was based on faulty command line input.
6	No resources found.

Examples

You can use the **lssensor** command without any parameters to obtain a complete list of sensor names.

1. To remove the sensor named NumLog, type:

```
rmsensor NumLog
```

Files

/opt/csm/bin/rmsensor	Location of the **rmsensor** command

See Also

The **chsensor**, **lssensor, mksensor**, **rmrsrc** commands.

IBM Cluster Systems Management for Linux Administration Guide for information on the ACL authorization file.

Author

Bruce Potter - cluster@us.ibm.com

rpower Command

Name

rpower – Controls remote power for Cluster Systems Management (CSM) nodes.

Synopsis

rpower [**–a**] [**−B** | **−P**] [**–h**] [**–n** *host[,host...]*] [**–N** *Node_group[,Node_group...]*] [**–v**] **on | off | reboot | query | resetsp**

Description

The **rpower** command allows remote power control for the 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. In addition to these options, you can define an environment variable, RPOWER_LIST, which is the name of 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 a response for each node that specifies the node name and the **on** or **off** status. The **on**, **off**, **reboot**, and **resetsp** options print the node name with the command specified and the return code of the command.

Options

−a	Runs the command on all of the nodes in the cluster.
−B	Allows nodes specified by the **−a**, **−n**, and **−N** options to be in both the PreManagedNode table/class and the ManagedNode table/class. This option cannot be used with the **−P** option. If no options are specified, nodes are selected by default from the ManagedNode table/class only.
−h	Writes usage information to standard output.
–n *host[,host...]*	Specifies a list of nodes on which to run the command.
–N *Node_group[,Node_group...]*	Specifies one or more node groups on which to run the command.
−P	Allows nodes to be specifed from the PreManagedNode table/class only. This option cannot be used with the **−B** option. If no options are specified, nodes are selected by default from the ManagedNode table/class only.
−v	Writes the verbose messages of the command to standard output.
on	Powers the node or nodes on.
off	Requests a shutdown and schedules a power off based on the ISP timer.
reboot	Reboots the power on the node or nodes.
query	Reports the power status of the node or nodes.
resetsp	Resets the service processor for the node or nodes.

Environment

RPOWER_LIST
Specifies a file that contains definitions of the set of hosts, one per line, that comprise the node list.

Exit Status

0 Command has run successfully.

1 Error occurred with one or more of the remote power commands.

Security

While the **rpower** command currently only supports the Netfinity hardware, the architecture is designed to allow future enhancements for other hardware support. For Netfinity hardware, the **rpower** command requires a **netfinity_power.config** file containing the ID and password for each service processor. See the **netfinity_power.config** file man page for more information.

Note: All systems are shipped with a default service processor ID of USERID and password of PASSW0RD (that's P-A-S-S-W-zero-R-D). You should change the ID and passwords and update the **netfinity_power.config** file to reflect your changes.

Examples

1. To query one node, type:

   ```
   rpower -n clsn04 query
   ```

   The output is:

   ```
   clsn04 on
   ```

2. To query all nodes, type:

   ```
   rpower -a query
   ```

   The output is similar to:

   ```
   clsn05.ppd.pok.ibm.com on
   clsn04.ppd.pok.ibm.com on
   clsn03.ppd.pok.ibm.com on
   clsn02.ppd.pok.ibm.com on
   clsn01.ppd.pok.ibm.com on
   ```

3. To power off a node, type:

   ```
   rpower -n clsn04 off
   ```

   The output is:

   ```
   clsn04 off complete rc=0
   ```

4. To power on a node, enter:

   ```
   rpower -n clsn04 on
   ```

5. To display a list of the nodes in a node group, type:

   ```
   nodegrp -p test
   ```

   The output is similar to:

   ```
   clsn01.ppd.pok.ibm.com
   clsn02.ppd.pok.ibm.com
   clsn05.ppd.pok.ibm.com
   ```

6. To query the power status of the nodes in a node group, type:

   ```
   rpower -N test query
   ```

   The output is similar to:

```
clsn01.ppd.pok.ibm.com on
clsn02.ppd.pok.ibm.com on
clsn05.ppd.pok.ibm.com on
```

Files

/etc/opt/csm/netfinity_power.config	Location of the **netfinity_power.config** file.
/opt/bin/rpower	Location of the **rpower** command.

See Also

The **lsnode** command

IBM Cluster Systems Management for Linux Remote Control Guide and Reference

Author

John Simpson – cluster@us.ibm.com

setupks Command

Name

setupks – Installs Kickstart if necessary, creates the Kickstart configuration file, and defines the nodes to be installed by means of Kickstart to the CSM database

Synopsis

setupks [**−h**] [**−v**] [**−n** *node_list*] [**−x | −p** *pkg_path*] [*Attr=value* [*Attr=value* ...]]

Description

The **setupks** command installs Red Hat Kickstart if it is not already installed, collects configuration information, and creates the Kickstart configuration file template (**kscfg.tmpl**) to contain the information that it has collected. From this template, a generated Kickstart configuration file is created for each node. The generated configuration file on the node is called **/tftpboot/ks71/***node_ip_address***-kickstart**. For more details on the template and the generated file, particularly how and when to modify them, see “kscfg.tmpl File” on page 42, and see the sample annotated **kscfg.tmpl** file in *IBM Cluster Systems Management for Linux Planning and Installation Guide*.

Note: For the current version of the template, see **/opt/csm/install/kscfg.tmpl** on your system.

By default, the **setupks** command works on all PreManagedNode objects that have their *InstallMethod* attribute set to **kickstart**. A list of ManagedNode or PreManagedNode objects, or both, can be specified on the command line, and **setupks** sets up the nodes in the list whose *InstallMethod* attribute is set to **kickstart**.

Specifically, **setupks** performs these tasks:

- Copies the RPMs from the Red Hat CD-ROMs or from the directories specified by **-p**. Prompts for CD-ROMs as needed.
- Starts, DHCP, NFS and ATFTP daemons if they are not already running.
- Sets up the **/etc/dhcpd.conf** file for MAC address collection and for full install of the nodes.
- Creates the files that are necessary for network boot into **/tftpboot**.
- Gets the MAC addresses of all specified nodes by running **getmacs**.
- Generates a Kickstart configuration file for each node.
- Generates a **firstboot** script for each node. This script is run when the node is first booted to its hard drive after the Kickstart install is complete. It runs the **makenode** command.

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:

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

The default value for the attributes is the value that those attributes have on the management server. These values populate the **/etc/dhpd.conf** file. The files necessary for a network boot are also created in **/tftpboot**.

Options

−h	Writes usage information to standard output.
−n *node_list*	Specifies a list of nodes to install

−p *pkg_path*	Specifies one or more directories, separated by colons, that contain copies of the Red Hat CD-ROMs (*p1*:*p2*:*p3*). The default is **/mnt/cdrom**.
−v	Writes the verbose messages of the command to standard output.
−x	Specifies to not copy Red Hat disks. This cannot be used with the **-p** option.

Examples

1. To set up three nodes with the same net mask, gateway, and nameserver attributes as the management server, type:

   ```
   setupks -nclsn01,clsn02,clsn03
   ```

2. To set up all of the premanaged nodes whose *InstallMethod* attribute is **kickstart** and specify values for the net mask, gateway, and nameserver attributes, type:

   ```
   setupks Netmask=255.255.255.0 Gateway=9.10.11.12 Nameservers=9.10.11.251,9.10.12.251
   ```

Files

/opt/bin/setupks	Location of the **setupks** program.
/opt/csm/install/kscfg.tmpl	Location of the Kickstart configuration file template.
/opt/csm/install/firstboot.tmpl	Location of the firstboot configuration file template.
/etc/dhcpd.conf	Location of the DHCP configuration file.
/tftpboot/pxelinuxcfg/HEX	Location of the **PXE** configuration file, where HEX represents the host IP address in hexadecimal format. This file is modified by the program during different phases of the installation.
/tftpboot/ks71/*node-ip-address***-kickstart**	Location of the Kickstart configuration file generated for the node.
/tftpboot/bin/*node-hostname***.firstboot**	Location of the firstboot script generated for the node.

See Also

The **addnode**, **definenode**, **getmacs**, **installms**, **installnode**, **makenode** commands.

- *IBM Cluster Systems Management for Linux Administration Guide* for information on selection string syntax.
- *IBM Cluster Systems Management for Linux Planning and Installation Guide* for more information on the installation processes.
- *IBM Cluster Systems Management for Linux Remote Control Guide and Reference* for information on hardware control.

Author

Sean Safron – cluster@us.ibm.com

whichdb Command

Name

whichdb - Displays or changes the underlying database that CSM uses.

Synopsis

whichdb [**−h**] [**−v**] [**−d | −i**] [**file | rmc**]

whichdb −D *dbd_connect_string*

Description

The **whichdb** command is normally not intended to be run by users or administrators. Rather it is used by installation routines to set the database that CSM uses to store its persistent information. When all of CSM is installed, **whichdb** is set to **rmc**. This gives CSM the event capabilities it needs to respond appropriately to changes in persistent information. When **whichdb** is run from the command line, it is run on the management server.

When only the **csm.core** RPM is installed and being used by a standalone package like **dsh** or some other tool, **whichdb** can be set to **file** to allow the persistent information to be stored in a set of files, for example, **/var/lib/csm/file_db**. Thus, **csm.core** does not have to specify RMC or any database as a prerequisite.

Note: If **whichdb** is set to **rmc**, then the syntax for the select string is described in the section ″Using Expressions″ in the *IBM Cluster Systems Management for Linux Administration Guide*. If **whichdb** is set to **file**, then normal SQL syntax is used. The commands that use a select string are: **chnode**, **lsnode**, and **nodegrp**.

If no arguments are specified, **whichdb** displays the currently set DBD.

Options

−d Displays the full DBD connect string for the specified **dbd**: **file** or **rmc**. If no DBD is specified, the DBD connect string of the currently set DBD is displayed.

−D *dbd_connect_string*
Sets the DBD connect string that **csm.core** uses explicitly. This string is used as the first input parameter to the **DBI->connect()** function.

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

−i Initializes the data base specified by **dbd** or the currently set DBD. This creates the tables with the necessary attributes (columns) for CSM.

−v Writes the command's verbose messages to standard output.

Examples

1. To display the database that is currently being used by CSM, type:
```
whichdb
```
2. To set CSM to use a flat file for its database, type:
```
whichdb file
```

Files

/opt/csm/bin/whichdb	Location of the **whichdb** command

See Also

The **chnode**, **createnode**, **lsnode**, **nodegrp**, **rmnode** commands.

Author

Bruce Potter - cluster@us.ibm.com

Chapter 2. ERRM Commands

chcondition Command

Name

chcondition – Changes any of the attributes of a defined condition.

Synopsis

chcondition [**−h**] [**−c** *New_condition*] [**−r** *Resource_class*] [**−e** *Event_expression*] [**−E** *Rearm_expression*] [**−d** *Event_description*] [**−D** *Rearm_description*] [**−n** *Node_name*[,*Node_name*...]] [**−s** ″*Selection_string*″] [**−S c | w | i**] [**−T**] [**−V**] *Condition*

Description

The **chcondition** command changes the attributes of a defined condition to the values supplied. If the name of the condition is changed using the **-c** option, any links with responses remain intact.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Condition The name of an existing condition.

Options

−c *New_condition*
The *New_condition* name is a character string that identifies the condition. If the name contains spaces, it must be enclosed in quotes. A name cannot consist of all spaces, be null, or contain imbedded single quotes.

−d *Event_description*
Specifies a user-supplied text that describes the event expression.

−D *Rearm_description*
Specifies a user-supplied text that describes the rearm expression.

−e *Event_expression*
Specifies an event expression. The event expression determines when an event occurs. It includes a dynamic attribute of the *Resource_class* with a mathematical comparison symbol (for example, >, <) and a constant. When this expression evaluates to TRUE, an event is generated.

−E *Rearm_expression*
Specifies a rearm expression. After the *Event_expression* has evaluated to TRUE and an event is generated, the rearm expression determines when monitoring for the *Event_expression* begins again. Typically, the rearm expression prevents multiple events from being generated for the same event evaluation. The rearm expression includes a dynamic attribute of the *Resource_class* with a mathematical comparison symbol (for example, >) and a constant.

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

−n *Node_name*
Specifies the host name for a node or list of host names for a list of nodes, separated by commas, where this condition is registered. The default is the local node. Use asterisk (″*″) to register the condition on all nodes defined in the ManagedNode resource class.

−r *Resource_class*
Specifies the resource class to be monitored by this condition. The **lsrsrcdef** command can be used to list the resource class names.

−s ″*Selection_string*″
Specifies a selection string that is applied to all of the *Resource_class* attributes to determine which resources should be monitored by the *Event_expression*. The default is to monitor all resources within the *Resource_class*. The selection string must be enclosed within double or single quotation marks. For information on how to specify selection strings, see ″Using Expressions″ in the chapter ″Using the Monitoring Application″ of the *IBM Cluster Systems Management for Linux Administration Guide*.

−S c | w | i
The severity of the event:

c	Critical
w	Warning
i	Informational

The default is **i**.

−T Writes the command's trace messages to standard error. For your software-service organization's use only.

−V Writes the command's verbose messages to standard output.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred that was based on faulty command line input.

Security

The user needs write permission for the IBM.Condition resource class to run **chcondition**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To change the condition name from ″FileSystem space used up″ to ″Watch FileSystem space″, type:
```
chcondition -c "Watch FileSystem space"  "FileSystem space used up"
```
2. To change a rearm expression and rearm description for a condition with the name ″tmp space used up″, type:
```
chcondition -E "PercentTotUsed < 80"\
 -D "Start monitoring tmp again after it is less than 80 percent full" \
"tmp space used up"
```

Files

/usr/sbin/rsct/bin/chcondition
Location of the **chcondition** command.

See Also

The **lscondition**, **lscondresp**, **mkcondition**, **rmcondition** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding Event Response operations and information on how to use expressions and selection strings.

Author

Joseph Chaky - cluster@us.ibm.com

chresponse Command

Name

chresponse – Adds or deletes the actions of a response or renames a response.

Synopsis

chresponse [**−h**] **−a −n** *Action* [**−d** *Days_of_week*[*,Days_of_week...*]] [**−t** *Time_of_day*[*,Time_of_day...*]] **−s** *Action_script* [**−r** *Return_code*] [**−e a | r | b**] [**−o**] [**−T**] [**−V**] *Response*

chresponse [**−h**] **−p −n** *Action* [**−T**] [**−V**] *Response*

chresponse [**−h**] **−c** *New_response* [**−T**] [**−V**] *Response*

Description

The **chresponse** command adds an action to a response or deletes an action from a response. A response must have at least one action defined to it. Actions define commands to be run when the response is used with a condition and the condition occurs. The **chresponse** command can also be used to rename a response.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Response	The name of the response to be changed.

Options

−a Adds the action specification to *Response*.

−c *New_response*

The *New_response* name is a character string that identifies the response. If the name contains spaces, it must be enclosed in quotes. A name cannot consist of all spaces, be null, or contain imbedded single quotes.

−d *Days_of_week*

Specifies the days of the week when the action being defined can be run. *Days_of_week* and *Time_of_day* together define the interval when the action can be run.

Enter the numbers of the days separated by a plus sign (+) or as a range of days separated by a hyphen (-). More than one *Days_of_week* parameter can be specified, but the parameters must be separated by a comma(,). The number of *Days_of_week* parameters specified must match the number of *Time_of_day* parameters specified. The default is all days. If no value is specified but a comma is entered, the default value is used. The numbers of the days follow:

1	Sunday
2	Monday
3	Tuesday
4	Wednesday
5	Thursday
6	Friday
7	Saturday

−e a | r | b
Specifies the type of event that causes the action being defined to run:

a	Event
r	Rearm event
b	Both event and rearm event

The default is event (**a** option).

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

−n *Action*
Specifies the name of the action. When the **−a** option is used, this is the name of the action being defined. When the **−p** option is used, this is the name of the action to be deleted. Action names must be unique within a response. Only one action can be defined at a time.

−o Directs all standard output from *Action_script* to the audit log. The default is not to keep standard output. Standard error is always directed to the audit log.

−p Deletes *Action* from Response.

−r *Return_code*
Specifies the expected return code for *Action_script*. The actual return code of *Action_script* is compared to the expected return code. A message is written to the audit log indicating whether they match. If the **−r** option is not specified, the actual return code is written to the audit log, and no comparison is performed.

−s *Action_script*
Specifies the fully qualified path for the script or command to run for the action being defined. See the man pages for **displayevent**, **logevent**, **msgevent**, **notifyevent**, and **wallevent** for descriptions of predefined response scripts that are provided with the application.

−t *Time_of_day*
Specifies the time range when *Action* can be run, consisting of the start time followed by the end time, separated by a hyphen. *Days_of_week* and *Time_of_day* together define the interval when the action can be run.

The time is in 24–hour format (HHMM) where the first two digits represent the hour and the last two digits represent the minutes. The start time must be less than the end time because the time is specified by day of the week. More than one *Time_of_day* parameter can be specified, but the parameters must be separated by a comma (,). The number of *Days_of_week* parameters specified matches the number of *Time_of_day* parameters specified. The default is 0000-2400. If no value is specified but a comma is entered, the default value is used.

−T Writes the command’s trace messages to standard error. For your software-service organization’s use only.

−V Writes the command’s verbose messages to standard output.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0 Command has run successfully.

1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred that was based on faulty command line input.

Security

The user needs write permission for the IBM.EventResponse resource class to run **chresponse**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To delete the action named ″E-mail root″ from the response named ″E-mail root anytime″ (″E-mail root″ cannot be the only action), type:

```
 chresponse -p -n "E-mail root"  "E-mail root anytime"
```

2. To add the action named ″E-mail root″ to be used Monday through Friday from 8 am to 6 pm that uses the command **/usr/sbin/rsct/bin/notifyevent root**, that saves standard output in the audit log, and that expects return code 5 from the action to the response ″E-mail root any time″, type:

```
chresponse  -a  -n "E-mail root" -d 2-6 -t 0800-1800 \
-s "/usr/sbin/rsct/bin/notifyevent root" -o  -r 5  \
"E-mail root anytime"
```

3. To rename the response ″E-mail root anytime″ to ″E-mail root and admin any time″, type:

```
chresponse  -c "E-mail root and admin anytime"   "E-mail root anytime"
```

Files

/usr/sbin/rsct/bin/chresponse
Location of the **chresponse** command.

See Also

The **lscondresp**, **lsresponse**, **mkcondresp**, **mkresponse**, **rmresponse** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding ERRM operations.

Author

Joseph Chaky - cluster@us.ibm.com

lscondition Command

Name

lscondition – Lists information about one or more conditions.

Synopsis

lscondition [**−h**] [**−A**] [**−m | −n | −e**] [**−C | −l | −t | −d | −D** *Delimiter*] [**−q**] [**−x**] [**−T**] [**−V**] [*Condition* [*Condition...*]]

Description

The **lscondition** command lists information about defined conditions.

The following information about the condition is listed:

EventDescription
The text description of *EventExpression*.

EventExpression
The expression used in monitoring this condition.

MonitorStatus
The status of the condition: monitored without error, not monitored, or in error.

Name The name of the condition.

NodeNames
The host names of the nodes where the condition is registered, if any. An asterisk (*) indicates all nodes defined in the ManagedNode resource class.

RearmDescription
The text description of *RearmExpression*.

RearmExpression
The expression used in determining when monitoring should restart for this condition after an event has occurred.

ResourceClass
The resource class monitored by this condition.

Severity
The severity of the condition: critical, warning, or informational.

SelectionString
The selection string that is applied to the attributes of the *ResourceClass* to determine which resources are included in the monitoring of this condition.

For a list of all conditions, enter the **lscondition** command without any condition names specified. A list of all the condition names is returned with the monitoring status for each condition. The default format in this case is tabular.

For all the information about all condition names, specify the **-A** option with the **lscondition** command. The **-A** option causes all information about a condition to be listed when no condition names are specified. When all the information about all conditions is listed, the default format is long. If a monitoring-status option (**-e**, **-m**, or **-n**) is specified, the conditions with that status are listed.

When more than one *Condition* is specified, the condition information is listed in the order that the condition names are entered.

By default, when a condition name is entered with the **lscondition** command, all of the attributes of the condition are displayed.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Condition The *Condition* may be a condition name or a substring of a condition name. When it is a substring, any defined condition name that contains the substring will be listed.

Options

−A Displays all of the attributes of the condition.

−C Outputs the **mkcondition** command that could be used to create the *Condition*. If more than one condition is specified, every **mkcondition** command appears on a separate line. This option is ignored when no conditions are specified. This option overrides the **–l** option.

−d Produces delimiter-formatted output. The default delimiter is a colon (:). Use the **–D** option if you wish to change the default delimiter.

−D *Delimiter*
Produces delimiter-formatted output that uses the specified delimiter. Use this option to specify something other than the default, colon (:). An example is when the data to be displayed contains colons. Use this option to specify a delimiter of one or more characters.

−e Lists only those conditions that are monitored in error.

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

−l Produces long formatted output. Displays the condition information on separate lines.

−m Lists only those conditions that are being monitored without error.

−n Lists only those conditions that are not being monitored.

−q Does not return an error when *Condition* does not exist.

−t Produces tabular formatted output. The condition information is displayed in separate columns.

−T Writes the command’s trace messages to standard error. For your software-service organization’s use only.

−V Writes the command’s verbose messages to standard output.

−x Suppresses header printing.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0 Command has run successfully.

1 Error occurred with RMC.

2 Error occurred with CLI script.

3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred that was based on faulty command line input.

Security

The user needs read permission for the IBM.Condition resource class to run **lscondition**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To list all conditions and their monitoring status, type:

   ```
   lscondition
   ```

   Output is similar to:

   ```
   Name                 MonitorStatus
   "FileSystem space used up"  "Monitored"
   "tmp space used up"  "Not monitored"
   "var space used up"  "Error"
   ```

2. To list general information about the condition ″FileSystem space used up″ in long form, type:

   ```
   lscondition "FileSystem space used up"
   ```

   Output is similar to:

   ```
      Name              = "FileSystem space used up"
      MonitorStatus     = "Monitored"
      ResourceClass     = "IBM.FileSystem"
      EventExpression   = "PercentTotUsed > 99"
      EventDescription  = "Generate event when space used is
   greater than 99 percent full"
      RearmExpression   = "PercentTotUsed < 85"
      RearmDescription  = "Start monitoring again after it is
   less than 85 percent"
      SelectionString   = ""
      Severity          = "w"
      NodeNames         = "localnode"
   ```

3. To list the command that would create the condition ″FileSystem space used up″, type:

   ```
   lscondition -C "FileSystem space used up"
   ```

   Output is similar to:

   ```
   mkcondition -r IBM.FileSystem -e "PercentTotUsed > 99"\
    -E "PercentTotUsed < 85"\
    -d "Generate event when space used is greater than 99 percent full"\
    -D "Start monitoring after it is less than 85 percent"\
    -S w "FileSystem space used up"
   ```

4. To list all conditions that have ″space″ in their name, type:

   ```
   lscondition space
   ```

   Output is similar to:

   ```
   Name = "FileSystem space used up"
   MonitorStatus = "Monitored"
   :
   Name = "tmp space used up"
   ```

```
MonitorStatus = "Not Monitored"
:
Name = "var space used up"
MonitorStatus = "Monitored"
```

Files

/usr/sbin/rsct/bin/lscondition
Location of the **lscondition** command.

See Also

The **chcondition**, **lscondresp**, **mkcondition**, **rmcondition** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding Event Response operations.

Author

Grant McLaughlin - cluster@us.ibm.com

lscondresp Command

Name

lscondresp – Lists information about a condition and its linked responses, if any.

Synopsis

lscondresp [**−h**] [**−a | −n**] [**−C | −l | −t | −d | −D** *Delimiter*] [**−q**] [**−x**] [**−T**] [**−V**] [*Condition* [*Response* [*Response ...*]]]

lscondresp [**−h**] [**−a | −n**] [**−C | −l | −t | −d | −D** *Delimiter*] [**−q**] [**−x**] **−r** [**−T**] [**−V**] *Response* [*Response ...*]

Description

The **lscondresp** command lists information about a condition and its linked responses. The information shows what responses are linked with a condition and whether monitoring is active for a condition and its linked response.

The following information is listed:

Condition	The name of the condition linked with a response.
State	State of the response for the condition. The state indicates whether a specified response is active or not.
Response	The name of the response linked with the condition.

To list a particular condition and response, specify both condition and response. To list all responses to a condition, specify only the condition. To list all conditions that a response is linked to, specify only the response along with the **-r** option. To list all conditions and their linked responses, omit specifying the condition and response. When neither the **−a** option nor the **−n** option is specified, all selected conditions for the responses are listed. Tabular format is the default.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Condition	The *Condition* may be a condition name or a substring of a condition name. When it is a substring, any defined condition name that contains the substring and is linked to the response will be listed.
Response	The *Response* may be a response name or a substring of a response. When it is a substring, any defined response name that contains the substring and is linked to the condition will be listed.

Options

−a
Lists only those responses that are active for the condition.

−C
Outputs the **mkcondition** command that could be used to create the *Condition*. If more than one condition is specified, every **mkcondition** command appears on a separate line. This option is ignored when no conditions are specified. This option overrides the **–l** option.

−d
Specifies delimiter-formatted output. The default delimiter is a colon (:). Use the **-D** option if you wish to change the default delimiter.

−D *Delimiter*
Specifies delimiter-formatted output that uses the specified delimiter. Use this option to specify

something other than the default, colon (:). For example, when the data to be displayed contains colons, use this option to specify another delimiter of one or more characters.

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

−l Specifies long formatted output. Displays the condition and response information on separate lines.

−n Lists only those responses that are not active for the condition.

−q Does not return an error if either *Condition* or *Response* does not exist.

−r Indicates that all command parameters are responses. There are no conditions specified. This lists all linked condition-response information for the specified responses.

−t Specifies tabular formatted output. The condition information and response information are displayed in separate columns.

−T Writes the command's trace messages to standard error. For your software-service organization's use only.

−V Writes the command's verbose messages to standard output.

−x Suppresses header printing.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0 Command has run successfully.

1 Error occurred with RMC.

2 Error occurred with CLI script.

3 Incorrect flag on command line.

4 Incorrect parameter on command line.

5 Error occurred that was based on faulty command line input.

Security

The user needs read permission for the IBM.Association resource class to run **lscondresp**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To list information about the condition ″FileSystem space used up″ with the linked response ″Broadcast event on-shift″, type:

   ```
   lscondresp  "FileSystem space used up" "Broadcast event on-shift"
   ```

   Output is similar to:

   ```
    Condition                    Response                        State
   "FileSystem space used up"   "Broadcast event on-shift"      "Active"
   ```

2. To list information about the condition ″FileSystem space used up″, type:

```
lscondresp "FileSystem space used up"
```

Output is similar to:

```
Condition                       Response                    State
"FileSystem space used up"     "Broadcast event on-shift"  "Active"
"FileSystem space used up"     "E-mail root anytime"       "Not Active"
```

3. To list information about the condition ″FileSystem space used up″ for responses that are active, type:

```
lscondresp -a  "FileSystem space used up"
```

Output is similar to:

```
Condition                       Response                    State
"FileSystem space used up"     "Broadcast event on-shift"  "Active"
```

4. To list all conditions with their linked responses, type:

```
lscondresp
```

Output is similar to:

```
Condition                       Response                    State
"FileSystem space used up"     "Broadcast event on-shift"  "Active"
"FileSystem space used up"     "E-mail root anytime"       "Not Active"
"Page in Rate"                 "Log event anytime"         "Active"
```

5. To list all conditions that have ″space″ in their name that are linked to a response, type:

```
lscondresp space
```

Output is similar to:

```
Condition                       Response                    State
"FileSystem space used up"     "Broadcast event on-shift"  "Active"
"FileSystem space used up"     "E-mail root anytime"       "Not Active"
```

Files

/usr/sbin/rsct/bin/lscondresp
Location of the **lscondresp** command.

See Also

The **mkcondition**, **mkcondresp**, **mkresponse**, **rmcondresp**, **startcondresp**, **stopcondresp** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding Event Response operations.

Author

Grant McLaughlin - cluster@us.ibm.com

lsresponse Command

Name

lsresponse – Lists information about one or more responses.

Synopsis

lsresponse [**−h**] [**−A**] [**−C | −l | −t | −d | −D** *Delimiter*] [**−q**] [**−x**] [**−T**] [**−V**] [*Response* [*Response...*]]

Description

The **lsresponse** command lists information about defined responses. The following information about the response is included:

Action

Name of an action.

ActionScript

The script or command to run for the action.

CheckReturnCode

Indicates if the actual return code for **ActionScript** is compared to its expected return code. The character ’y’ is for yes and ’n’ is for no.

DaysOfWeek

The days of the week when the action is allowed to be run. *Days_of_week* and *Time_of_day* together define the interval when the action can be run.

The numbers of the days can be separated by a plus sign (+) or displayed as a range of days separated by a hyphen (-). If there is more than one *Days_of_week*, they are separated by a comma (,). The number of *Days_of_week* matches the number of *Time_of_day*. The numbers of the days follow:

1	Sunday
2	Monday
3	Tuesday
4	Wednesday
5	Thursday
6	Friday
7	Saturday

EventType

The type of event that causes the action to be run; event, rearm event, or both.

Name Name of the response.

ReturnCode

The expected return code for **ActionScript**.

StandardOut

Indicates if standard output is directed to the audit log. The character ’y’ is for yes and ’n’ is for no.

TimeOfDay

The time range when *Action* can be run, consisting of the start time followed by the end time separated by a hyphen. *Days_of_week* and *Time_of_day* together define the interval when the action can be run.

The time is in 24–hour format (HHMM) where the first two digits represent the hour and the last two digits represent the minutes. The start time must be less than the end time because the time is specified by day of the week. If there is more than one *Time_of_day*, they are separated by a comma (,). The number of *Days_of_week* matches the number of *Time_of_day*.

To get a list of all response names, enter the **lsresponse** command alone without any response names specified. A list of all response names is returned. The default format in this case is tabular. To see all the information about all response names, specify the **-A** option with the **lsresponse** command. The **-A** option causes all information about a response to be listed when no response names are specified. When all the information about all responses is listed, the long format is the default.

When more than one *Response* is specified, the response information is listed in the order that the responses are entered.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Response The *Response* may be a response name or a substring of a response name. When it is a substring, any defined response name that contains the substring will be listed.

Options

−A Displays all of the attributes of the response.

−C Outputs the **mkresponse** command that can be used to create the response and one of its actions. If more than one response is specified, every **mkresponse** command appears on a separate line. This option is ignored when no responses are specified. This option overrides the **–l** option.

−d Specifies delimiter-formatted output. The default delimiter is a colon (:). Use the **–D** option if you wish to change the default delimiter.

−D *Delimiter*
Specifies delimiter-formatted output that uses the specified delimiter. Use this option to specify something other than the default, colon (:). For example, when the data to be displayed contains colons, use this option to specify another delimiter of one or more characters.

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

−l Specifies long formatted output. Displays the response information on separate lines.

−q Does not return an error when *Response* does not exist.

−t Specifies tabular formatted output. The response information is displayed in separate columns.

−T Writes the command’s trace messages to standard error. For your software-service organization’s use only.

−V Writes the command’s verbose messages to standard output.

−x Suppresses headers when printing.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred that was based on faulty command line input.

Security

The user needs read permission for the IBM.EventResponse resource class to run **lsresponse**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To list general information about the response ″Critical notifications″, type:

   ```
   lsresponse "Critical notifications"
   ```

   Output is similar to:

   ```
   Name             = "Critical notifications"
   Action           = "Log Critical Event"
   DaysOfWeek       = 1+2+7
   TimeOfDay        = 0000-2400
   ActionScript     = "/usr/sbin/rsct/bin/logevent /tmp/criticalEvents"
   ReturnCode       = 0
   CheckReturnCode  = "y"
   EventType        = "b"
   StandardOut      = "y"

   Name             = "Critical notifications"
   Action           = "E-mail root"
   DaysOfWeek       = 6+2,6+2,6+5
   TimeOfDay        = 1700-2400,0000-0800,0000-2400
   ActionScript     = "/usr/sbin/rsct/bin/notifyevent root"
   ReturnCode       = 0
   CheckReturnCode  = "y"
   EventType        = "b"
   StandardOut      = "y"
   ```

2. To list all of the responses, type:

   ```
   lsresponse
   ```

   Output is similar to:

   ```
   Name
   "E-mail root anytime"
   "E-mail root first shift"
   "Critical notifications"
   ```

3. To list the command that would create the response ″Critical notifications″ along with one of its actions, type:

   ```
   lsresponse -C "Critical notifications"
   ```

   Output is similar to:

   ```
   mkresponse -n "Log Critical Event" -d 1+2+7 -t 0000-2400 \
   -s "usr/sbin/rsct/bin/logevent /tmp/criticalEvents"  \
   -e b -r 0 "Critical notifications"
   ```

4. To list all responses that have ″E-mail″ in their name, type:

   ```
   lsresponse "E-mail"
   ```

   Output is similar to:

   ```
   Name = "E-mail root anytime"
   Action = "some name"
   :
   Name = "E-mail root first shift"
   Action = "some name"
   ```

Files

/usr/sbin/rsct/bin/lsresponse
Location of the **lsresponse** command.

See Also

The **chresponse**, **lscondresp**, **mkcondresp**, **mkresponse**, **rmresponse** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding Event Response operations.

Author

Grant McLaughlin - cluster@us.ibm.com

mkcondition Command

Name

mkcondition – Creates a new condition definition which can be monitored.

Synopsis

mkcondition [**−h**] **−r** *Resource_class* **−e** *Event_expression* [**−E** *Rearm_expression*] [**−d** *Event_description*] [**−D** *Rearm_description*] [**−n** *Node_name*[,*Node_name*...]] [**−s** *"Selection_string"*] [**−S c | w | i**] [**−T**] [**−V**] *Condition*

mkcondition [**−h**] **−c** *Existing_condition* [**−r** *Resource_class*] [**−e** *Event_expression*] [**−E** *Rearm_expression*] [**−d** *Event_description*] [**−D** *Rearm_description*] [**−n** *Node_name*[,*Node_name*...]] [**−s** *"Selection_string"*] [**−S c | w | i**] [**−T**] [**−V**] *Condition*

Description

The **mkcondition** command creates a new condition with the name specified by the *Condition* parameter. The condition is used to monitor a resource for the occurrence of the condition (or event). One or more responses to an event can be defined by using the **mkresponse** command. The conditions can then be linked to the responses by using the **mkcondresp** command, or by using the **startcondresp** command to link responses and start monitoring.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Condition The *Condition* name is a character string that identifies the condition. If the name contains spaces, it must be enclosed in quotes. A name cannot consist of all spaces, be null, or contain imbedded single quotes.

Options

−c *Existing_condition*
Copy an existing condition. If any other options are specified, update the new condition as indicated by the options. Links with responses are not copied.

−d *Event_description*
Specifies user-supplied text that describes the event expression.

−D *Rearm_description*
Specifies user-supplied text that describes the rearm expression.

−e *Event_expression*
Specifies the event expression, which determines when an event occurs. It includes a dynamic attribute of the *Resource_class* with a mathematical comparison symbol (for example, >, <) and a constant. When this expression evaluates to TRUE, an event is generated.

−E *Rearm_expression*
Specifies the rearm expression. After the *Event_expression* has evaluated to TRUE and an event is generated, the rearm expression determines when monitoring for the *Event_expression* begins again. Typically, the rearm expression prevents multiple events from being generated for the same event evaluation. The rearm expression includes a dynamic attribute of the *Resource_class* with a mathematical comparison symbol (for example, >) and a constant.

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

−n *Node_name*
Specifies the host name for a node or list of host names for a list of nodes, separated by commas, where this condition is registered. The default is the local node. Use an asterisk (″*″) to register the condition on all nodes defined in the ManagedNode resource class.

−r *Resource_class*
Specifies the resource class to be monitored by this condition. The **lsrsrcdef** command can be used to list the resource-class names.

−s ″*Selection_string*″
Specifies a selection string that is applied to all of the *Resource_class* attributes to determine which resources should be monitored by the *Event_expression*. The default is to monitor all resources within the *Resource_class*. The selection string must be enclosed within double or single quotation marks. For information on how to specify selection strings, see ″Using Expressions″ in the chapter ″Using the Monitoring Application″ of the *IBM Cluster Systems Management for Linux Administration Guide*.

−S c | w | i
Specifies the severity of the event:

c	Critical
w	Warning
i	Informational

The default is **i**. This is for user reference.

−T Writes the command’s trace messages to standard error. For your software-service organization’s use only.

−V Writes the command’s verbose messages to standard output.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred that was based on faulty command line input.

Security

The user needs write permission for the IBM.Condition resource class to run **mkcondition**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To define a condition with the name ″FileSystem space used up″ to check for percentage of space used greater than 90% and to rearm when the percentage is back down below 85%, type:

   ```
   mkcondition -r IBM.FileSystem  \
   -e "PercentTotUsed > 90" -E "PercentTotUsed < 85" \
   "FileSystem space used up"
   ```

2. To define a condition with the name ″tmp space used up″ to check for percentage of space used greater than 90% for **/tmp** and to rearm when the percentage is back down below 85%, including comments, type:

   ```
   mkcondition -r IBM.FileSystem  \
   -e "PercentTotUsed > 90" -E "PercentTotUsed < 85" \
   -d "Generate event when tmp > 90% full"  \
   -D "Restart monitoring tmp again after back down < 85% full"\
   -s 'Name=="/tmp"'  "tmp space used up"
   ```

3. To define a condition with the name ″Space used up″ as a copy of ″FileSystem space used up″, type:

   ```
   mkcondition -c "FileSystem space used up"  "Space used up"
   ```

4. To define a condition with the name ″var space used up″ as a copy of ″tmp space used up″, but change the selection to **/var**, type:

   ```
   mkcondition -c "tmp space used up" -s 'Name=="/var"' \
   "var space used up"
   ```

Files

/usr/sbin/rsct/bin/mkcondition
Location of the **mkcondition** command.

See Also

The **chcondition**, **lscondition**, **mkcondresp**, **mkresponse**, **rmcondition**, **startcondresp** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding Event Response operations and information on how to use expressions and selection strings.

Author

Joseph Chaky - cluster@us.ibm.com

mkcondresp Command

Name

mkcondresp – Creates a link between a condition and one or more responses.

Synopsis

mkcondresp [**−h**] [**−T**] [**−V**] *Condition Response* [*Response...*]

Description

The **mkcondresp** command creates a link between a condition and a response, but does not start monitoring. More than one response may be linked with a condition. You can start monitoring for this condition and its linked responses later by using the **startcondresp** command.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Condition	Specifies the name of the condition to be linked to the response. The condition is always specified first.
Response	Specifies the name of the response, or more than one response. If more than one response is given, all responses are linked to the condition.

Options

−h	Writes the command’s usage statement to standard output.
−T	Writes the command’s trace messages to standard error. For your software-service organization’s use only.
−V	Writes the command’s verbose messages to standard output.

Environment

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred that was based on faulty command line input.

Security

The user needs write permission for the IBM.Association resource class to run **mkcondresp**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To link the condition ″FileSystem space used″ to the response ″Broadcast event on-shift″, type:
   ```
   mkcondresp "FileSystem space used" "Broadcast event on-shift"
   ```
2. To link the condition ″FileSystem space used″ to the responses ″Broadcast event on-shift″ and ″E-mail root anytime″, type:
   ```
   mkcondresp "FileSystem space used" "Broadcast event on-shift" "E-mail root anytime"
   ```

Files

/usr/sbin/rsct/bin/mkcondresp
Location of the **mkcondresp** command.

See Also

The **lscondresp**, **mkcondition mkresponse**, **rmcondresp**, **startcondresp**, **stopcondresp** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding ERRM operations.

Author

Joseph Chaky - cluster@us.ibm.com

mkresponse Command

Name

mkresponse – Creates a new response definition with one action.

Synopsis

mkresponse [**−h**] **−n** *Action* [**−d** *Days_of_week*[*,Days_of_week...*]] [**−t** *Time_of_day*[*,Time_of_day...*]] **−s** *Action_script* [**−r** *Return_code*] [**−e a | r | b**] [**−o**] [**−T**] [**−V**] *Response*

mkresponse [**−h**] **−c** *Existing_response* [**−T**] [**−V**] *Response*

Description

The **mkresponse** command creates a new response definition with the name specified by the *Response* parameter. One action must also be defined when the response is defined. Actions define commands to be run when the response is used with a condition and the condition occurs. The action defines days of the week when the action can be used, the time of day for those days of the week, the script or command to be run, what type of event causes the command to be run, the expected return code of the script or command, and whether to keep standard output. The days and times are paired so that different times can be specified for different days.

The **chresponse** command can be used to add actions to a response or to remove actions from a response. A response must always have at least one action defined. Monitoring can be started by using the **startcondresp** command. The **startcondresp** command links a response to a condition if they are not already linked.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Response
The *Response* name is a character string that identifies the response. If the name contains spaces, it must be enclosed in quotes. A name cannot consist of all spaces, be null, or contain imbedded single quotes.

Options

−c *Existing_response*
Copies an existing response. Links with conditions are not copied.

−d *Days_of_week*

Specifies the days of the week when the action being defined can be run. *Days_of_week* and *Time_of_day* together define the interval when the action can be run.

Enter the numbers of the days separated by a plus sign (+) or as a range of days separated by a hyphen (-). More than one *Days_of_week* parameter can be specified, but the parameters must be separated by a comma (,). The number of *Days_of_week* parameters specified must match the number of *Time_of_day* parameters specified. The default is all days. If no value is specified but a comma is entered, the default value is used. The numbers of the days follow:

1	Sunday
2	Monday
3	Tuesday
4	Wednesday

5	Thursday
6	Friday
7	Saturday

−e a | r | b
Specifies the type of event that causes the action being defined to run:

a	Event
r	Rearm event
b	Both event and rearm event

The default is event (**a** flag).

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

−n *Action*
Specifies the name of the action being defined. Only one action can be defined when the response is created. Use the **chresponse** command to add more actions to the response.

−o Directs all standard output from *Action_script* to the audit log. The default is not to keep standard output. Standard error is always directed to the audit log.

−r *Return_code*
Specifies the expected return code for *Action_script*. If the expected return code is specified, the actual return code of *Action_script* is compared to the expected return code. A message is written to the audit log indicating whether they match. If the **−r** option is not specified, the actual return code is written to the audit log, and no comparison is performed.

−s *Action_script*
Specifies the fully qualified path for the script or command to run for the action being defined. See the man pages for **displayevent**, **logevent**, **msgevent**, **notifyevent**, and **wallevent** for descriptions of the predefined response scripts provided with the application.

−t *Time_of_day*
Specifies the time range when *Action* can be run, consisting of the start time followed by the end time, separated by a hyphen. *Days_of_week* and *Time_of_day* together define the interval when the action can be run.

The time is in 24−hour format (HHMM) where the first two digits represent the hour and the last two digits represent the minutes. The start time must be less than the end time because the time is specified by day of the week. More than one *Time_of_day* parameter can be specified, but the parameters must be separated by a comma (,). The number of *Days_of_week* parameters specified must match the number of *Time_of_day* parameters specified. The default value is 0000-2400. If no value is specified but a comma is entered, the default value is used.

−T Writes the command's trace messages to standard error. For your software-service organization's use only.

−V Writes the command's verbose messages to standard output.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0 Command has run successfully.

1 Error occurred with RMC.

2 Error occurred with CLI script.

3 Incorrect flag on command line.

4 Incorrect parameter on command line.

5 Error occurred that was based on faulty command line input.

Security

The user needs write permission for the IBM.EventResponse resource class to run **mkresponse**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To define a response with the name ″E-mail root anytime″ that has an action named ″E-mail root″, to be used anytime Saturday and Sunday and uses the command **/usr/sbin/rsct/bin/notifyevent root** for both events and rearm events, type:

```
 mkresponse -n "E-mail root" -d 1+7  \
-s "/usr/sbin/rsct/bin/notifyevent root" -e b   \
"E-mail root anytime"
```

2. To define a response with the name ″E-mail root anytime″ that has an action named ″E-mail root″, to be used anytime Saturday and Sunday but only 8 am to 5 pm Monday through Friday and that uses the command **/usr/sbin/rsct/bin/notifyevent root** for events, type:

```
mkresponse -n "E-mail root"     \
-d 1+7,2-6  -t 0000-2400,0800-1700  \
-s "/usr/sbin/rsct/bin/notifyevent root" -e a  \
"E-mail root anytime"
```

3. To define a response with the name ″E-mail root first shift″ that has an action named ″E-mail root″ to be used Monday through Friday from 8 am to 6 pm, that uses the command **/usr/sbin/rsct/bin/notifyevent root** for rearm events, and that saves standard output in the audit log, expecting return code 5, type:

```
mkresponse -n "E-mail root" -d 2-6  -t 0800-1800  \
-s "/usr/sbin/rsct/bin/notifyevent root" -e r  -o   \
-r 5  "E-mail root first shift"
```

4. To define a response with the name ″Critical notifications″ as a copy of ″Warning notifications″, type:

```
mkresponse -c "Warning notifications" "Critical notifications"
```

Files

/usr/sbin/rsct/bin/mkresponse
Location of the **mkresponse** command.

See Also

The **chresponse**, **lsresponse**, **mkcondition**, **mkcondresp**, **rmresponse**, **startcondresp** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding ERRM operations.

Author

Joseph Chaky - cluster@us.ibm.com

rmcondition Command

Name

rmcondition – Removes a condition.

Synopsis

rmcondition [**−h**] [**−f**] [**−q**] [**−T**] [**−V**] *Condition*

Description

The **rmcondition** command removes the condition specified by the *Condition* parameter. The condition must already exist to be removed. When the condition must be removed even if it has linked responses, use the **-f** option to force the condition and the links with the responses to be removed. If the **-f** option is not specified and links with responses exist, the condition is not removed. Responses are not removed by this command.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Condition	The name of a condition to be removed.

Options

−f	Forces the condition to be removed even if it is linked to responses. The links with the responses are removed as well as the condition, but the responses are not removed.
−h	Writes the command's usage statement to standard output.
−q	Does not return an error when *Condition* does not exist.
−T	Writes the command's trace messages to standard error. For your software-service organization's use only.
−V	Writes the command's verbose messages to standard output.

Environment

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred that was based on faulty command line input.

Security

The user needs write permission for the IBM.Condition resource class to run **rmcondition**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To remove the condition definition named ″FileSystem space used up″, type:
   ```
   rmcondition "FileSystem space used up"
   ```
2. To remove the condition definition named ″FileSystem space used up″ even if the condition is linked with responses, type:
   ```
   rmcondition -f "FileSystem space used up"
   ```

Files

/usr/sbin/rsct/bin/rmcondition
Location of the **rmcondition** command.

See Also

The **chcondition**, **lscondition**, **lscondresp**, **mkcondition** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding ERRM operations.

Author

Joseph Chaky - cluster@us.ibm.com

rmcondresp Command

Name

rmcondresp – Deletes a link between a condition and one or more responses.

Synopsis

rmcondresp [**−h**] [**−q**] [**−T**] [**−V**] *Condition* [*Response* [*Response* ...]]

rmcondresp [**−h**] [**−q**] **−r** [**−T**] [**−V**] *Response* [*Response* ...]

Description

The **rmcondresp** command deletes the link between a condition and a response. More than one response can be specified. The response is no longer run when the condition occurs. Use the **-r** option to specify that the command parameters consist only of responses. This deletes all links to conditions for these responses. If only a condition is specified, links to all responses for that condition are deleted.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Condition	Specifies the name of the condition linked to the response. The condition is always specified first unless the **-r** option is used.
Response	Specifies the name of a response or more than one response. The links from the specified responses to the specified condition are removed.

Options

−h	Writes the command’s usage statement to standard output.
−q	Does not return an error when either *Condition* or *Response* does not exist.
−r	Indicates that all command parameters are responses. There are no conditions specified. This command removes links from all conditions that are linked to the specified responses.
−T	Writes the command’s trace messages to standard error. For your software-service organization’s use only.
−V	Writes the command’s verbose messages to standard output.

Environment

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.

4 Incorrect parameter on command line.

5 Error occurred that was based on faulty command line input.

Security

The user needs write permission for the IBM.Association resource class to run **rmcondresp**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To delete the link between the condition ″FileSystem space used″ and the response ″Broadcast event on-shift″, type:
   ```
   rmcondresp  "FileSystem space used" "Broadcast event on-shift"
   ```
2. To delete the links between the condition ″FileSystem space used″ and all of its responses, type:
   ```
   rmcondresp  "FileSystem space used"
   ```
3. To delete the links between the condition ″FileSystem space used″ and the responses ″Broadcast event on-shift″ and ″E-mail root anytime″, type:
   ```
   rmcondresp  "FileSystem space used"  "Broadcast event on-shift" "E-mail root anytime"
   ```
4. To delete the links between the response ″Broadcast event on-shift″ and all of the conditions that use it, type:
   ```
   rmcondresp -r "Broadcast event on-shift"
   ```

Files

/usr/sbin/rsct/bin/rmcondresp
Location of the **rmcondresp** command.

See Also

The **lscondresp**, **mkcondition**, **mkcondresp**, **mkresponse**, **startcondresp**, **stopcondresp** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding ERRM operations.

Author

Joseph Chaky - cluster@us.ibm.com

rmresponse Command

Name

rmresponse – Removes a response.

Synopsis

rmresponse [**−h**] [**−f**] [**−q**] [**−T**] [**−V**] *Response*

Description

The **rmresponse** command removes the response specified by the *Response* parameter. The response must already exist in order to be removed. When the response must be removed even if it is linked with conditions, specify the **-f** flag. This forces the response and the links with the conditions to be removed. If the **-f** option is not specified and links with conditions exist, the response is not removed. Conditions are not removed by this command.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Response	The name of a defined response to be removed.

Options

−f	Forces the response to be removed even if it is linked with conditions. The links with the conditions are removed as well as the response, but the conditions are not removed.
−h	Writes the command’s usage statement to standard output.
−q	Does not return an error when *Response* does not exist.
−T	Writes the command’s trace messages to standard error. For your software-service organization’s use only.
−V	Writes the command’s verbose messages to standard output.

Environment

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred that was based on faulty command line input.

Security

The user needs write permission for the IBM.EventResponse resource class to run **rmresponse**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To remove the response definition named ″Broadcast event on-shift″, type:

   ```
    rmresponse  "Broadcast event on-shift"
   ```
2. To remove the response definition named ″Broadcast event on-shift″ even if the response is linked with conditions, type:

   ```
   rmresponse -f "Broadcast event on-shift"
   ```

Files

/usr/sbin/rsct/bin/rmresponse
Location of the **rmresponse** command.

See Also

The **chresponse**, **lscondresp**, **lsresponse**, **mkcondresp**, **mkresponse** commands.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding ERRM operations.

Author

Joseph Chaky - cluster@us.ibm.com

startcondresp Command

Name

startcondresp – Starts monitoring a condition that has one or more linked responses.

Synopsis

startcondresp [**−h**] [**−T**] [**−V**] *Condition* [*Response* [*Response ...*]]

Description

The **startcondresp** command starts the monitoring of a condition that has a linked response. After monitoring is started, when the condition occurs, the response is run. If no responses are specified, monitoring is started for all responses linked to the condition. This causes all of the linked responses to run when the condition occurs. If more than one response is specified, monitoring is started only for those linked responses.

If one or more responses are specified and the responses are not linked with the condition, the **startcondresp** command links the specified responses to the condition, and monitoring is started. Use the **mkcondresp** command to link a response to a condition without starting monitoring.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Condition	Specifies the name of the condition linked to the response. The condition is always specified first.
Response	Specifies the name of a response or more than one response. Specifying more than one response links the responses to the condition if they are not already linked and starts monitoring for the specified responses.

Options

−h	Writes the command's usage statement to standard output.
−T	Writes the command's trace messages to standard error. For your software-service organization's use only.
−V	Writes the command's verbose messages to standard output.

Environment

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.

4	Incorrect parameter on command line.
5	Error occurred that was based on faulty command line input.

Security

The user needs write permission for the IBM.Association resource class to run **startcondresp**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To start monitoring for the condition ″FileSystem space used ″ by using the response ″Broadcast event on-shift″, whether or not the response is linked with the condition, type:
   ```
   startcondresp  "FileSystem space used" "Broadcast event on-shift"
   ```
2. To start monitoring for the condition ″FileSystem space used ″ by using all of its linked responses, type:
   ```
   startcondresp "FileSystem space used"
   ```
3. To start monitoring for the condition ″FileSystem space used ″ by using the response ″Broadcast event on-shift″ and ″E-mail root anytime″, whether or not they are linked with the condition, type:
   ```
   startcondresp "FileSystem space used" "Broadcast event on-shift" "E-mail root anytime"
   ```

Files

/usr/sbin/rsct/bin/startcondresp
Location of the **startcondresp** command.

See Also

The **lscondresp**, **mkcondition**, **mkcondresp**, **mkresponse**, **stopcondresp** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding Event Response operations.

Author

Joseph Chaky - cluster@us.ibm.com

stopcondresp Command

Name

stopcondresp – Stops monitoring a condition that has one or more linked responses.

Synopsis

stopcondresp [**−h**] [**−q**] [**−T**] [**−V**] *Condition* [*Response* [*Response ...*]]

Description

The **stopcondresp** command stops the monitoring of a condition that has a linked response. If no response is specified, all of the linked responses for the condition are stopped. If more than one response is specified, only the linked responses are stopped. When the condition occurs, the response is not run. If no responses are active for a condition, the condition is no longer monitored.

Note: The ERRM commands are normally run on the management server for central monitoring. However, they can also be run on individual nodes for monitoring conditions and running responses locally on the nodes. The CT_CONTACT environment variable also affects whether a command can be run on a node.

Condition	Specifies the name of the condition linked to the response. The condition is always specified first.
Response	Specifies the names of one or more responses. Monitoring is stopped for the specified responses. (If a specified response is not linked to the condition, it is ignored.)

Options

−h	Writes the command’s usage statement to standard output.
−q	Does not return an error when either *Condition* or *Response* does not exist or when the *Condition* linked with *Response* is not being monitored.
−T	Writes the command’s trace messages to standard error. For your software-service organization’s use only.
−V	Writes the command’s verbose messages to standard output.

Environment

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred that was based on faulty command line input.

Security

The user needs write permission for the IBM.Association resource class to run **stopcondresp**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To stop monitoring for the condition ″FileSystem space used ″ which has the response ″Broadcast event on-shift″ linked with it, type:

   ```
   stopcondresp "FileSystem space used" "Broadcast event on-shift"
   ```
2. To stop monitoring for the condition ″FileSystem space used ″ using all of its linked responses, type:

   ```
   stopcondresp  "FileSystem space used"
   ```

Files

/usr/sbin/rsct/bin/stopcondresp
Location of the **stopcondresp** command.

See Also

The **lscondresp**, **mkcondition**, **mkcondresp**, **mkresponse**, **startcondresp** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains information regarding ERRM operations.

Author

Joseph Chaky - cluster@us.ibm.com

Chapter 3. RMC Commands

rmccli General Information File

Name

rmccli – Contains information global to the Resource Monitoring and Control (RMC) command-line interface (CLI).

Description

This man page provides global information for the Resource Monitoring and Control command-line interface, including data types, terminology and references to other related material.

Terminology

Common terminology used in the RMC CLI man pages:

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 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.

Resource

A resource is an instance of a resource class. Therefore, a resource contains instances of the persistent and dynamic attributes for the resource class. A resource of the class IBM.FileSystem contains the persistent and dynamic attributes that describe a particular file system.

Resource_class

A resource class is a collection of attributes that describe similar hardware or software entities. The IBM.FileSystem resource class, for example, consists of persistent and dynamic attributes that define existing file systems in a host. To see all of the resource classes defined in the system, issue the command **lsrsrc** without any options or parameters. To see all of the IBM.FileSystem resources defined in the system, issue the command: **lsrsrc IBM.FileSystem**.

Selection_string

Specifies a selection string. All selection strings must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
```

```
-s 'Name ?= "test"'
```

Only persistent attributes may be listed in a selection string. For information on how to specify selection strings, see ″Using Expressions″ in the chapter ″Using the Monitoring Application″ of the *IBM Cluster Systems Management for Linux Administration Guide*.

Data display information

The options that control the display function for the RMC CLI routines, in order of precedence, are:

1. **–l** for long display. This is the default display format.

   For example, the command:

   ```
   lsrsrc -s 'Name == "c175n05"' IBM.Foo Name NodeList SD Binary RH Int32Array
   ```

   produces output similar to:

   ```
   Persistent Attributes for Resource: IBM.Foo
   resource 1:
           Name       = "c175n05"
           NodeList   = {1}
   ```

```
SD         = ["testing 1 2 3",1,{0,1,2}]
Binary     = "0xaabbcc00 0xeeff"
RH         = "0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000"
Int32Array = {1,5,-10,1000000}
```

2. **–t** for tabular display.

   For example, the command:

   ```
   lsrsrc -s 'Name ?= "Page"' -t IBM.Condition Name EventExpression
   ```

   produces output similar to:

   ```
   Persistent Attributes for Resource: IBM.Condition

   Name                  EventExpression
   "Page space out rate" "VMPgSpOutRate > 500"
   "Page fault rate"     "VMPgFaultRate > 500"
   "Page out rate"       "VMPgOutRate > 500"
   "Page in rate"        "VMPgInRate > 500"
   "Page space in rate"  "VMPgSpInRate > 500"
   ```
3. **–x** for suppressing headers when printing.
4. **–d** for colon (:) delimited display.

   For example, the command:

   ```
   lsrsrc -xd -s 'Name == "c175n05"' IBM.Foo Name Int32 Uint32Array SD Binary
   ```

   produces output similar to:

   ```
   c175n05:-100:{}:["hel  lo1",1,{0,1,2}]:"0xaabbcc00 0xeeff":
   ```

   Note the use of the **–x** option along with the **–d** option.
5. **–D** *Delimiter* for string-delimited display.

   For example, the command:

   ```
   lsrsrc -xD:: -s 'Name == "c175n05"' IBM.Foo Name Int32 Uint32Array SD Binary
   ```

   produces output similar to:

   ```
   c175n05::-100::{}::["hel  lo1",1,{0,1,2}]::"0xaabbcc00 0xeeff"::
   ```

   Note the use of the **–x** option along with the **–D** *Delimiter* option.

When output of any list command (**lsrsrc**, **lsrsrcdef**) is displayed in the tabular output format, the printing column width may be truncated. If more characters need to be displayed (as in the case of strings) use the **–l** option to display the entire field.

Data input formatting

Binary data may be input in the following formats:

- ″0x######## 0x######## 0x####...″
- ″0x###################...″
- 0x################...

Be careful when you specify strings as input data:

- Strings that contain no white space or non-alphanumeric characters may be supplied as input without enclosing quotation marks.
- Strings that contain white space or other alphanumeric characters must be enclosed in double quotation marks.
- Strings that contain single quotation marks (’) must be enclosed by double quotation marks (″), as shown in this example: `"this is a string with 'single quotations marks'"`

Selection strings must be input in double quotation marks, unless the selection string itself contains double quotation marks, in which case the selection string must be enclosed in single quotation marks. For information on how to specify selection strings, see ″Using Expressions″ in the chapter ″Using the Monitoring Application″ of the *IBM Cluster Systems Management for Linux Administration Guide*.

- Typical selection string input: `"NodeNumber == 1"`.
- Selection string input where double quotation marks are part of the selection string: `'Name == "c175n05"'`.

Structured data (SD) types must be enclosed in square brackets: `[hello,1,{2,4,6,8}]`

When supplying structured data (SD) as command-line input to the RMC commands, enclose the SD in single quotation marks: `SD='[hello,1,{2,4,6,8}]'`

Arrays of any type must be enclosed in braces {}:

- Array of integers: `{-4, -3, -2, -1, 0, 1, 2, 3, 4}`
- Array of strings: `{abc, "do re mi", 123}`
- Array of structured data: `{[hello,1,{0,1,2,3}],[hello2,2,{2,4,6,8}]}`

Arrays of any type, with more than one element, must be enclosed in quotes. For example:

- mkrsrc IBM.Foo Name=testing NodeList={1} Uint32Array=’{1,2,3}’
- mkrsrc IBM.Foo Name=testing NodeList=’{1}’ Uint32_array=’{1,2,3}’

Arrays of strings and arrays of structured data must always be enclosed in quotes.

When supplying arrays of structured data or arrays containing strings enclosed in quotation marks as command-line input to the RMC commands, enclose the entire array in single quotation marks:

- Array of strings: `mkrsrc IBM.Foo Name="c175n05" NodeList={1} StringArray='{"a string","a different string"}'`
- Array of structured data: `mkrsrc IBM.Foo Name="c175n05" NodeList={1} SDArray='{["string 1",1,{1,1}],["string 2",2,{1,2,3}]}'`

For more examples, refer to the **Resource_Data_Input** file.

Data output formatting

String data is always displayed in either double or single quotation marks, as shown below:

- A description attribute that equals the string ″This is a string that contains white space″ is displayed using long format as:
  ```
  Description = "This is a string that contains white space"
  ```
- A description attribute value that equals an empty string ″″ is displayed in long format as:
  ```
  Description = ""
  ```
- A description attribute value that equals a string that contains a new-line character at the end of the string is displayed in long format as:
  ```
  Description = "This string ends with a new - line character...
   "
  ```
- A selection string containing double quotation marks is displayed in long format as:
  ```
  SelectionString = 'Name == "c175n05"'
  ```
- A name attribute value that equals the string ″c175n05″ is displayed in long format as:
  ```
  Name = "c175n05"
  ```

Binary data is displayed as follows:

```
"0x######## 0x######## 0x######## 0x###..."
```

Naming and numbering conventions

The following keywords are used throughout the RMC command man pages:

Attr The name of a resource class or resource attribute.

Resource_class
The name of a resource class.

Command structure and use

The RMC commands may be grouped into categories representing the different operations that can be performed on resource classes and resources:

- Creating and removing: **mkrsrc**, **rmrsrc**
- Modifying: **chrsrc**, **refrsrc**
- Viewing definitions and data: **lsrsrc**, **lsrsrcdef**
- Viewing actions: **lsactdef**

The RMC commands can be run directly from the command line or called by user-written scripts. In addition, the RMC commands are used as the basis for higher-level commands, such as the Event Response Resource Manager (ERRM) command line interface.

Options

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

−T Writes this command's trace messages to standard error. For your software-service organization's use only.

−V Writes this command's verbose messages to standard output.

All RMC commands support a **-V** and **-T** option. The **-V** option is used to see additional information (verbose mode) regarding the command. Verbose messages are contained in message catalogs and are translated based on the locale in which you are running and other criteria.

Run a command with the **-T** option only when the your software-service organization instructs you to turn on trace. Trace messages are not translated. The **-T** option shows the calls and returns to and from the underlying Perl to C Extensions.

Environment

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

CT_LOCAL_SCOPE

The CT_LOCAL_SCOPE environment variable sets the RMC subsystem scope to the stand-alone environment. By default, the RMC subsystem monitors and controls the resources and resource classes on all nodes in the cluster. A cluster can consist of one or more nodes. When a cluster consists of one node, it is said to operate in a stand-alone environment. The scope of a command in the stand-alone environment is the local node. The scope of a command in a cluster environment of more than one node is the entire cluster. For a cluster of more than one node, to return or change resources or resource classes only on the node where RMC is running, set the CT_LOCAL_SCOPE environment variable to 1, as follows:

```
export CT_LOCAL_SCOPE=1
```

For example, assume that a cluster consists of nodes A, B, and C and the following command is run:

```
lsrsrc IBM.FileSystem
```

Resources for the file systems associated with nodes A, B, and C are returned.

If CT_LOCAL_SCOPE is set to 1 and the **lsrsrc** command is run on Node A, then only the file system resources associated with node A are returned.

See Also

The **Resource_Data_Input** file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding RMC operations.

Author

Stephanie Beals - cluster@us.ibm.com

Resource_Data_Input File

Name

Resource_Data_Input File – Input file for passing both resource class and resource attribute names and values to the Resource Monitoring and Control command-line interface (CLI). The data in this file is used for defining resources or for changing persistent attribute values of a resource or resource class.

Description

The **Resource_Data_Input File** is used in conjunction with the **-f** command line option to pass resource persistent attribute values to the RMC CLI in cases where using the command line directly would be too cumbersome or too prone to typographical errors. This file has no set location. It can be a temporary or permanent file, depending on requirements.

The **mkrsrc** and **chrsrc** commands read this file when they are issued with the **-f** option. The **lsrsrcdef** and **lsactdef** commands generate a file with this format when issued with the **-i** option.

PersistentResourceAttributes
Persistent attribute names and values for one or more resources for a specific resource class used to define a new resource or change attribute values for an existing resource. The persistent resource attributes are read in by the commands **mkrsrc** and **chrsrc**. These attributes are ignored if the input file is read by the **chrsrc** command that has been specified with the **-c** option.

PersistentResourceClassAttributes
Persistent attribute names and values for a resource class used to change the attribute values of an existing resource class. The persistent resource class attributes are read in by the command **chrsrc** only when the **-c** option is specified.

In general, the **Resource_Data_Input File** is a flat text file with a format as follows (bold words are literal). Note that text preceding any single colon is an arbitrary label and may be any alphanumeric text.

```
PersistentResourceAttributes::
# This is a comment
    label:
      AttrName1  = value
      AttrName2  = value
      AttrName3  = value
    another label:
        Name        = name
        NodeNumber  = 1
...
::

PersistentResourceClassAttributes::
    label:
      SomeSettableAttrName  = value
      SomeOtherSettableAttrName  = value
    ::
...
```

See the **Examples** section for more details.

Format points:

- The keywords: **PersistentResourceAttributes** and **PersistentResourceClassAttributes** are all followed by a double colon (::).
- The order of the keyword stanzas is not significant in the file. For example, **PersistentResourceClassAttributes** could precede **PersistentResourceClass**. It does not affect the portion of the data that is read in by the calling CLI.

- Individual stanza headings (beneath the keywords) are followed by a single colon (:). Example: `c175n05 resource info:`
- White space at the beginning of lines is not significant. Tabs or spaces are suggested for readability.
- Any line whose first printable character is a pound sign (#) is a comment.
- Each entry on an individual line is separated by white space (spaces or tabs).
- Blank lines in the file are not significant and are suggested for readability.
- There is no limit to the number of resource attribute stanzas included in a particular **PersistentResourceAttributes** section.
- There is no limit to the number of resource class attribute stanzas included in a particular **PersistentResourceClassAttributes** section. Typically, there is only one instance of a resource class. In this case, only one stanza is expected.
- If only one resource attribute stanza is included in a particular **PersistentResourceAttributes** section, the `label:` keyword can be omitted.
- If only one resource class attribute stanza is included in a particular **PersistentResourceClassAttributes** section, the `label:` keyword can be omitted.
- Values that contain spaces must be enclosed in double quotation marks or single quotation marks.
- A double colon (::) indicates the end of a section. If a terminating double colon is not found, the next Reserved Keyword or end of file signals the end of a section.
- Double quotation marks included within a string that is surrounded by double quotation marks must be escaped. (\″).

  Note: Double quotation marks can be nested within single quotation marks.
  These are examples:
  - `"Name == \"testing\""`
  - `'Name == "testing"'`

    This syntax is preferred if your string is a selection string and you are going to cut and paste to the command line.
- Single quotation marks included within a string that is surrounded by single quotation marks must be escaped. (\').

  Note: Single quotation marks can be nested within double quotation marks.
  These are examples:
  - `'Isn\'t that true'`
  - `"Isn't that true"`

    This syntax is preferred if you are going to cut and paste to the command line.
- The format you use to enter data in the **Resource Data Input** file may not be the same format used on the command line. The shell you choose to run the commands in has its own rules with regard to quotation marks. Refer to the documentation for your shell for these rules, which determine how to enter data on the command line.

Examples

1. Example of input file for IBM.Foo resource used by the following sample **mkrsrc** command. Comments are arbitrary:

   ```
   mkrsrc -f /tmp/my_resource_data_input_file IBM.Foo
   ```

   The file consists of:

   ```
   PersistentResourceAttributes::
   # Resource 1 - only set required attributes
   resource 1:
       Name="c175n04"
   ```

```
    NodeList = {1}
# Resource 2 - setting both required and optional attributes
# mkrsrc -e2 IBM.Foo displays required and optional
# persistent attributes
resource 2:
    Name="c175n05"
    NodeList = {1}
    Int32 = -99
    Uint32 = 99
    Int64 = -123456789123456789
    Uint64 = 123456789123456789
    Float32 = -9.89
    Float64 = 123456789.123456789
    String = "testing 123"
    Binary = 0xaabbccddeeff
    RH = "0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000"
    SD = [hello,1,{2,4,6,8}]
    Int32Array = {-4, -3, -2, -1, 0, 1, 2, 3, 4}
    Int64Array = {-4,-3,-2,-1,0,1,2,3,4}
    Uint32Array = {0,1,2,3,4,5,6}
    Uint64Array = {0,1,2,3,4,5,6}
    Float32Array = {-3.3, -2.2, -1.2, 0, 1, 2.2, 3.3}
    Float64Array = {-3.3, -2.2, -1.2, 0, 1, 2.2, 3.3}
    StringArray = {abc,"do re mi", 123}
    BinaryArray = {"0x01", "0x02", "0x0304"}
    RHArray     = {"0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000",
 "0xaaaa 0xaaaa 0xbbbbbbbb 0xcccccccc 0xdddddddd 0xeeeeeeee"}
    SDArray     = {[hello,1,{0,1,2,3}],[hello2,2,{2,4,6,8}]}
```

2. Example of an input file for changing the attribute values of existing IBM.Foo resources.

   ```
   chrsrc -f  /tmp/Foo/ch_resources -s 'Name == "c175n05"' IBM.Foo
   ```

   The file consists of:

   ```
   PersistentResourceAttributes::
   # Changing resources that match the selection string entered
   # when running chrsrc command.
   resource 1:
           String           = "this is a string test"
           Int32Array       = {10,-20,30,-40,50,-60}
   ```

See Also

The **chrsrc**, **lsactdef**, **lsrsrcdef**, **mkrsrc** commands.

The **rmccli** General Information file.

Author

Stephanie Beals - cluster@us.ibm.com

chrsrc Command

Name

chrsrc – Changes the persistent attribute values of a resource or resource class.

Synopsis

To change the persistent attribute values of a resource or resource class (**-c** option), using data entered on the command line:

chrsrc [**−h**] **−s** ″*Selection_string*″ [**−v**] [**−T**] [**−V**] *Resource_class Attr=value*...

chrsrc [**−h**] **−r** [**−v**] [**−T**] [**−V**] *Resource_handle Attr=value*...

chrsrc [**−h**] **−c** [**−v**] [**−T**] [**−V**] *Resource_class Attr=value*...

To change the persistent attribute values of a resource or resource class (**-c** option), using data predefined in an input file:

chrsrc [**−h**] **−f** *Resource_Data_Input_file* **−s** ″*Selection_string*″ [**−v**] [**−T**] [**−V**] *Resource_class*

chrsrc [**−h**] **−f** *Resource_Data_Input_file* **−r** [**−v**] [**−T**] [**−V**] *Resource_handle*

chrsrc [**−h**] **−f** *Resource_Data_Input_file* **−c** [**−v**] [**−T**] [**−V**] *Resource_class*

Description

The **chrsrc** command changes the persistent attribute values for a resource or resource class.

By default, this command changes the resource persistent attributes. If you wish to change the resource class persistent attributes, specify the **-c** option.

Only persistent attributes that do not have a ″Read Only″ property can be changed. The underlying resource manager that controls the specified resource determines which attributes can be changed by using this command.

Use the **-v** option to verify that the attribute names specified by the command line or **Resource_Data_Input** file are valid persistent attributes for the specified resource and that these attributes do not have a property of ″Read Only″. When **chrsrc** is run with the **-v** option, the specified attributes are not changed.

After the **chrsrc** command is run without the **-v** option, if any specified attribute could not be changed, it is noted in an error message. If a particular attribute passes when the **chrsrc -v** command is run, this does not ensure that the attribute can be changed. The underlying resource manager controls which attributes can be changed.

To change the persistent attributes for a resource, use the **-r** option to change just the resource linked with the specified *Resource_handle*, or use the **-s** option to change all the resources that match the specified ″*Selection_string*″.

Attr=value

Enter one or more attribute value pairs. If the **-f** option is specified, *Attr=value* pair parameters should not be entered on the command line. *Attr* is any defined persistent attribute name. Use the **lsrsrcdef** command to get a list of the defined persistent attributes and their data types for the specified resource. The value entered must be the appropriate data type for the specified attribute. For example, if NodeNumber is defined as a uint32 data type, enter a positive numeric value.

Resource_class
A resource class name. Enter the **lsrsrcdef** command for a list of defined resource class names.

Resource_handle
The specific resource handle that is linked with the resource that you wish to change. Use the **lsrsrc** command to obtain valid resource handles. The resource handle must be enclosed within double quotation marks. An example is: `"0x4017 0x0001 0x00000000 0x0069684c 0x0d4715b0 0xe9635f69"`.

Options

−c Changes the resource class persistent attribute values. By default, the resource persistent attributes are changed. Use this option if you want to change the resource class persistent attributes.

−f *Resource_Data_Input_file*
Specifies the name of the file which contains resource attribute information.

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

−r Changes the specific resource that matches the specified *Resource_handle*.

−s ″*Selection_string*″
Specifies a selection string. All selection strings must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'

-s 'Name ?= "test"'
```

Only persistent attributes may be listed in a selection string. For information on how to specify selection strings, see ″Using Expressions″ in the chapter ″Using the Monitoring Application″ of the *IBM Cluster Systems Management for Linux Administration Guide*.

−T Writes the command’s trace messages to standard error. For your software-service organization’s use only.

−v Verifies that all of the specified attributes are valid persistent attribute names that do not have a Read Only property. The **chrsrc** command cannot verify that all of the attributes specified on the command line or in the input file can be changed. The underlying resource manager that controls the specified resource determines which attributes can be changed. After this command is run (without the **-v** option), any attribute name that could not be changed is included in an error message.

−V Writes the command’s verbose messages to standard output.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0 Command has run successfully.

1 Error occurred with RMC.

2 Error occurred with CLI script.

3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred with RMC that was based on faulty command line input.
6	No resources were found that match the selection string.

Security

The user needs write permission for the *Resource_class* specified in **chrsrc** to run **chrsrc**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To change the Int32, Uint32 and SD persistent resource attributes in resource class IBM.Foo for the resources that have a Name equal to c175n05, type:

   ```
   chrsrc -s 'Name == "c175n05"' IBM.Foo \
   Int32=-9999 Uint32=9999\
   SD='["testing 1 2 3",1,{2,4,6}]'
   ```

2. To change the Int32, Uint32, SD persistent resource attributes in resource class IBM.Foo for the resource that has a Name starting with `c175n`, using the **Resource_Data_Input** file with the following contents:

   ```
   PersistentResourceAttributes::
   resource 1:
        Int32  = -9999
        Uint32 = 9999
        SD     = ["testing 1 2 3",1,{2,4,6}]
   ```

   type:

   ```
   chrsrc -f /tmp/IBM.Foo.chrsrc\
    -s 'Name ?= "c175n"'  IBM.Foo
   ```

3. To change the Name persistent resource attribute for the resource that has a resource handle equal to ″0x0001 0x4005 0x35ae868c 0x00000000 0xfeef2948 0x0d80b827″, type:

   ```
   chrsrc -r "0x0001 0x4005 0x35ae868c 0x00000000 0xfeef2948 0x0d80b827" Name="c175n05"
   ```

Files

/usr/sbin/rsct/bin/chrsrc	Location of the **chrsrc** command.

See Also

The **lsrsrc**, **lsrsrcdef**, **mkrsrc**, **rmrsrc** commands.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding RMC operations and information on how to use expressions and selection strings.

Author

Stephanie Beals - cluster@us.ibm.com

lsactdef Command

Name

lsactdef – Lists (displays) action definitions of a resource or resource class.

Synopsis

lsactdef [**−h**] [**−c**] [**−p** *Property*] [**−s i | o**] [**−e**] [**−l | −i | −t | −d | −D** *Delimiter*] [**−x**] [**−T**] [**−V**] [*Resource_class* [*Action*...]]

Description

The **lsactdef** command lists the action definitions of a resource or resource class. Use **lsactdef** with no parameters specified to get a list of all resource class names.

By default, the resource action definitions are displayed. To see the resource class action definitions, specify the **-c** option.

By default, when no actions are specified on the command line, only actions that are defined as public are displayed. To override this default, use the **-p** option or specify on the command line the names of the actions whose definition you wish to display.

To see the structured data definition that is required as input when this action is invoked, specify the **-s i** option. To see the structured data definition linked with the output that results from invoking this action, specify the **-s o** option.

By default, for the actions that contain descriptions, the descriptions are not displayed. Specify the **-e** option to display the descriptions. Because the translation of these descriptions may take some time and because some of the descriptions are very long, the default is not to display these descriptions.

Action If a *Resource_class* parameter is specified, zero or more action names may be specified. If no *Action* parameter is specified, the definitions for all of the actions for the *Resource_class* are listed. Specific action names may be entered to control which actions are displayed and in what order. Actions must be separated by spaces.

Resource_class
Resource class name. The name of the resource class whose action definitions you wish displayed. If no *Resource_class* parameter is specified, a list of all of the resource class names is displayed.

Options

−c Displays the resource class action definitions. By default the resource action definitions are displayed.

−d Specifies delimiter-formatted output. The default delimiter is a colon (:). Use the **-D** option if you wish to change the default delimiter.

−D *Delimiter*
Specifies delimiter-formatted output that uses the specified delimiter. Use this option to specify a delimiter other than the default colon (:). An example is when the data to be displayed contains colons. Use this option to specify a delimiter of one or more characters.

−e Expands the descriptions. By default the definitions are displayed without the textual descriptions because they can be lengthy. Specify this option to see both the definitions and descriptions.

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

−i Specifies input format. Generates a template of the **Resource_Data_Input** file. The output is displayed in long (stanza) format. The attribute's SD element data types are displayed as the value in the *Attr=value* pairs. It is suggested that when you use this option, the output of the **lsactdef** command be directed to a file. This option overrides the **−s o** option.

−l Specifies long formatted output, one entry per line. This is the default display format. If the **lsactdef** command is issued with the **-l** option, but without a resource class name, the **-l** option is ignored when the command returns the list of defined resource class names.

p *Property*
Displays actions with the specified *Property*. By default, only the definitions for public actions are displayed. To display all action definitions regardless of the action property, use the **-p 0** option.

Action Properties

0x0001	Long running
0x0002	Public

A decimal or hexadecimal value can be specified for the property. To request the action definitions for all actions that have one or more properties, ″OR″ the properties of interest together and then specify the ″ORed″ value with the **-p** option. For example, to request the action definitions for all actions that are long running or public, type:

```
-p 0x03
```

−s i | o
Displays the structured data definition for the action input or action response.

i Displays the action input structured data definitions. This is the default.

o Displays the action response (output) structured data definitions.

−t Specifies tabular formatted output. Each attribute is displayed in a separate column, one resource per line.

−T Writes the command's trace messages to standard error. For your software-service organization's use only.

−V Writes the command's verbose messages to standard output.

−x Suppresses header printing.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0 Command has run successfully.

1 Error occurred with RMC.

2 Error occurred with CLI script.

3 Incorrect flag on command line.

4 Incorrect parameter on command line.

5 Error occurred with RMC that was based on faulty command line input.

Security

The user needs read permission for the *Resource_class* specified in **lsactdef** to run **lsactdef**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To list the names of all of the resource classes, type:

   ```
   lsactdef
   ```

   Output is similar to:

   ```
   class_name
   "IBM.Association"
   "IBM.AuditLog"
   "IBM.AuditLogTemplate"
   "IBM.Condition"
   "IBM.EventResponse"
   "IBM.Host"
   "IBM.Program"
   "IBM.Sensor"
   "IBM.ManagedNode"
   ...
   ```

2. To list the public resource action definitions for resource class IBM.AuditLog, type:

   ```
   lsactdef IBM.AuditLog
   ```

   Output is similar to:

   ```
   Resource Action Definitions for
   class_name: IBM.AuditLog
   action 1:
           action_name    = "GetRecords"
           display_name   = ""
           description    = ""
           properties     = {"public"}
           confirm_prompt = ""
           action_id      = 0
           variety_list   = {{1..1}}
           variety_count  = 1
           timeout        = 0
   action 2:
           action_name    = "DeleteRecords"
           display_name   = ""
           description    = ""
           properties     = {"public"}
           confirm_prompt = ""
           action_id      = 1
           variety_list   = {{1..1}}
           variety_count  = 1
           timeout        = 0
   ....
   ```

3. To list the structured data definition required for invoking the action on resources in resource class IBM.AuditLog, action GetRecords, type:

   ```
   lsactdef -s i IBM.AuditLog GetRecords
   ```

   Output is similar to:

   ```
   Resource Action Input for: IBM.AuditLog
   action_name GetRecords:
   SD element 1:
           element_index     = 0
           element_name      = "MatchCriteria"
           description       = ""
   ```

```
        element_data_type = "char_ptr"
        display_name      = ""
SD element 2:
        element_index     = 1
        element_name      = "IncludeDetail"
        description       = ""
        element_data_type = "uint32"
        display_name      = ""
```

Files

/usr/sbin/rsct/bin/lsactdef	Location of the **lsactdef** command.

See Also

The **lsrsrcdef** command.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding RMC operations.

Author

Stephanie Beals - cluster@us.ibm.com

lsrsrc Command

Name

lsrsrc – Lists (displays) resources or a resource class.

Synopsis

lsrsrc [**−h**] [**−s** ″*Selection_string*″] [**−a p | d | b**] [**−p** *Property*] [**−l | −t | −d | −D** *Delimiter*] [**−x**] [**−T**] [**−V**] [*Resource_class* [*Attr*...]]

lsrsrc [**−h**] [**−s** ″*Selection_string*″] **−r** [**−l | −t | −d | −D** *Delimiter*] [**−x**] [**−T**] [**−V**] [*Resource_class*]

lsrsrc [**−h**] **−c** [**−a p | d | b**] [**−p** *Property*] [**−l | −t | −d | −D** *Delimiter*] [**−x**] [**−T**] [**−V**] *Resource_class* [*Attr*...]

Description

The **lsrsrc** command is used to list the persistent and dynamic attributes and their values of either a resource class or a resource.

When no *Attr* parameter is specified, only attributes that are defined as public are displayed. Use the **-p** option to override this default. When no *Attr* parameter is specified, the **-a p | d | b** option controls whether only persistent, or only dynamic, or both persistent and dynamic attributes and their values are displayed.

When one or more attribute names are specified, exactly the attribute names specified and their values are displayed in the order specified, provided that each of the specified attribute names are valid.

To get a list of all of the resource classes, enter the **lsrsrc** command with no parameters.

Specify the **-c** option to display a list of the resource class attributes and values.

Specify the **-r** option to display only the resource handles linked with the resources in the displayed class.

By default, the resource attributes and values are displayed in long format. Use the **-t**, **-d**, or **-D** option for the resources to be displayed in tabular or delimiter formatted output.

For best performance, specify either the **-a p** option or only persistent attributes as parameters.

Note: Any attribute that has a data type defined as ct_none (for example, a Quantum) is not listed by the **lsrsrc** command. RMC does not return attribute values for attributes that are defined as Quantum. To list attribute definitions, use the **lsrsrcdef** command.

Attr

Attribute name. Both persistent and dynamic attribute names may be specified to control which attributes are displayed and their order. Zero or more attributes may be specified. Attributes must be separated by spaces. If no attribute names are specified, the **-a p | d | b** option controls whether only persistent attributes, only dynamic attributes, or both persistent and dynamic attributes are displayed. When no attribute names are specified, only attributes that are defined as public are displayed. Use the **-p** option to override this default.

Resource_class

Resource class name. The name of the resource class whose resources you wish displayed. Zero or one *Resource_class* parameter can be specified. If no *Resource_class* parameter is specified, a list of all resource class names is displayed.

Options

−a p | d | b
Specifies an attribute type. By default only persistent attributes are displayed. This option can be used only when no attribute names are specified on the command line.

p	Displays only persistent attributes.
d	Displays only dynamic attributes.
b	Displays both persistent and dynamic attributes.

For best performance, specify the **-a p** option.

−c Displays the attributes for the resource class. By default, the resource attributes, not the resource class, are displayed. This option overrides the **-r** option.

−d Specifies delimiter-formatted output. The default delimiter is a colon (:). Use the **-D** option if you wish to change the default delimiter.

−D *Delimiter*
Specifies delimiter-formatted output that uses the specified delimiter. Use this option to specify something other than the default colon (:). An example is when the data to be displayed contains colons. Use this option to specify a delimiter of one or more characters.

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

−l Specifies long formatted output. Each attribute is displayed on a separate line. This is the default display format. If the **lsrsrc** command is issued with the **-l** option, but without a resource class name, the **-l** option is ignored when the command returns the list of defined resource class names.

−p *Property*
Displays attributes with the specified *Property*. By default, only public attributes are displayed. To display all the attributes regardless of the property, use the **-p 0** option. Use this option in conjunction with the **-a** option when no attributes are specified on the command line.

Persistent Attribute Properties

0x0001	Read only
0x0002	Required for define
0x0004	Not valid for define
0x0008	Optional for define
0x0010	Selectable
0x0020	Public

Dynamic Attribute Properties

0x0020	Public

A decimal or hexadecimal value can be specified for the property. To display attributes and their values for all attributes that have one or more properties, ″OR″ the properties of interest together and then specify the ″ORed″ value with the **-p** option. For example, to display attributes and their values for all persistent attributes that are either required for define or optional for define, type:

```
-p 0x0a
```

−r Displays the resource handles for the resources that match the specified selection string or all resources when no selection string is specified.

−s ″*Selection_string*″
Specifies a selection string. All selection strings must be enclosed within either double or single

quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
```

```
-s 'Name ?= "test"'
```

Only persistent attributes may be listed in a selection string. For information on how to specify selection strings, see ″Using Expressions″ in the chapter ″Using the Monitoring Application″ of the *IBM Cluster Systems Management for Linux Administration Guide*.

−t Specifies tabular formatted output. Each attribute is displayed in a separate column, one resource per line.

−T Writes the command’s trace messages to standard error. For your software-service organization’s use only.

−V Writes the command’s verbose messages to standard output.

−x Suppresses header printing.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0 Command has run successfully.

1 Error occurred with RMC.

2 Error occurred with CLI script.

3 Incorrect flag on command line.

4 Incorrect parameter on command line.

5 Error occurred with RMC that was based on faulty command line input.

Security

The user needs read permission for the *Resource_class* specified in **lsrsrc** to run **lsrsrc**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To list the names of all of the resource classes, type:

   ```
   lsrsrc
   ```

   Output is similar to:

   ```
   class_name
   "IBM.Association"
   "IBM.Condition"
   'IBM.EventResponse"
   ```

```
"IBM.Host"
"IBM.Ethernet"
"IBM.TokenRing"
...
```

2. To list the persistent attributes for resource IBM.Host that have 4 processors, type:

   ```
   lsrsrc -s "NumProcessors == 4" -a p -p 0 IBM.Host
   ```

   Output is similar to:

   ```
   Resource Persistent Attributes for: IBM.Host
   resource 1:
           Name           = "c175n05.ppd.pok.ibm.com"
           ResourceHandle = "0x4008 0x0001 0x00000000 0x0069684c 0x0d7f55d5 0x0c32fde3"
           Variety        = 1
           NodeList       = {1}
           NumProcessors  = 4
           RealMemSize    = 1073696768
   ```
3. To list the public dynamic attributes for resource IBM.Host on node 1, type:

   ```
   lsrsrc -s 'Name == "c175n05.ppd.pok.ibm.com"' -a d IBM.Host
   ```

   Output is similar to:

   ```
   Resource Dynamic Attributes for: IBM.Host
   resource 1:
           ProcRunQueue        = 1.03347987093142
           ProcSwapQueue       = 1.00548852941929
           TotalPgSpSize       = 65536
           TotalPgSpFree       = 65131
           PctTotalPgSpUsed    = 0.61798095703125
           PctTotalPgSpFree    = 99.3820190429688
           PctTotalTimeIdle    = 0
           PctTotalTimeWait    = 51.5244382399734
           PctTotalTimeUser    = 12.8246006482343
           PctTotalTimeKernel  = 35.6509611117922
           PctRealMemFree      = 66
           PctRealMemPinned    = 4
           RealMemFramesFree   = 173361
           VMPgInRate          = 0
           VMPgOutRate         = 0
           VMPgFaultRate       = 0
            ...
   ```
4. To list the Name, Variety, and ProcessorType attributes for the IBM.Processor resource on all the online nodes, type:

   ```
   lsrsrc IBM.Processor Name Variety ProcessorType
   ```

   Output is similar to:

   ```
   Resource Persistent Attributes for: IBM.Processor
   resource 1:
           Name          = "proc3"
           Variety       = 1
           ProcessorType = "PowerPC_604"
   resource 2:
           Name          = "proc2"
           Variety       = 1
           ProcessorType = "PowerPC_604"
   resource 3:
           Name          = "proc1"
           Variety       = 1
           ProcessorType = "PowerPC_604"
   resource 4:
           Name          = "proc0"
           Variety       = 1
           ProcessorType = "PowerPC_604"
   ```

5. To list both the persistent and dynamic attributes for the resource class IBM.Condition, type:

   ```
   lsrsrc -c -a b -p 0 IBM.Condition
   ```

   Output is similar to:

   ```
   Resource Class Persistent and Dynamic Attributes for: IBM.Condition
   resource 1:
           ResourceType = 0
           Variety      = 0
   ```

Files

/usr/sbin/rsct/bin/lsrsrc	Location of the **lsrsrc** command.

See Also

The **lsrsrcdef** command.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding RMC operations.

Author

Stephanie Beals - cluster@us.ibm.com

lsrsrcdef Command

Name

lsrsrcdef – Lists a resource or resource class definition.

Synopsis

lsrsrcdef [**−h**] [**−c**] [**−a p | d**] [**−p** *Property*] [**−e**] [**−s**] [**−l | −i | −t | −d | −D** *Delimiter*] [**−x**] [**−T**] [**−V**] [*Resource_class* [*Attr*...]]

Description

The **lsrsrcdef** command lists the definition of a resource class, or the persistent or dynamic attributes of a resource instance or a resource class.

Use **lsrsrcdef** with no parameters specified to get a list of all resource class names.

To see just the resource class definition, specify the **-c** option and a *Resource_class* parameter without specifying the **-a** option.

To see the persistent or dynamic attribute definitions of the resource class, specify a *Resource_class* parameter and the **-c** option with the appropriate **-a** option (**-a d** for dynamic attribute or **-a p** for persistent attribute definitions). To see the persistent or dynamic attribute definitions for a resource, specify the appropriate **-a p | d** option without the **-c** option.

By default, when no *Attr* parameters are specified on the command line, only the definitions for public attributes are displayed. To override this default, use the **-p** option or specify the name of the attribute you wish to display.

By default, for the attributes that contain descriptions, the descriptions are not displayed. Specify the **-e** option to display the descriptions. Because some of the descriptions are very long, the default is not to display them.

Attr If a *Resource_class* parameter is specified, zero or more attribute names can be specified. If no *Attr* parameter is specified, the definition for all the attributes for the resource are listed. Specific attribute names may be specified to control which attributes are displayed and their order. Specify only persistent attribute names when the **-a p** option is used. Specify only dynamic attribute names when the **-a d** option is used. Attributes must be separated by spaces.

Resource_class
Resource class name. The name of the resource class whose attribute definitions are to be displayed. If no *Resource_class* parameter is specified, a list of all of the resource class names is displayed.

Options

−a p | d
Specifies the attribute type. Either persistent or dynamic attribute definitions may be displayed. Use this option with the **-c** option for the persistent or dynamic attribute definitions of the resource class.

p Displays only persistent attributes

d Displays only dynamic attributes

−c Displays the resource class definition. By default, the resource definition is displayed. Combine with the **-a p | d** option to display the resource class definition for persistent or dynamic attributes. Use the **–c** option alone, without the **-a p | d** option, to see the resource class definition.

−d Specifies delimiter-formatted output. The default delimiter is a colon (:). Use the **-D** option to change the default delimiter.

−D *Delimiter*
Specifies delimiter-formatted output that uses the specified delimiter. Use this option to specify something other than the default colon (:). An example is when the data to be displayed contains colons. Use this option to specify a delimiter of one or more characters.

−e Expands the descriptions. By default the definitions are displayed, but the textual descriptions are not displayed because they may be lengthy. Specify this option to see both the definitions and descriptions.

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

−i Generates a template of the **Resource_Data_Input** file which can then, after appropriate editing, be used as input to the **mkrsrc** command. The output is displayed in long (stanza) format. All required and optional attributes that can be used to define a resource are displayed. The attribute data type is displayed as the value in the *Attr=value* pairs. It is suggested that when you use this option, the output of the **lsrsrcdef** command be directed to a file. This option overrides the **−s** and **−a d** options.

−l Specifies long formatted output, one entry per line. This is the default display format. If the **lsrsrcdef** command is issued with the **-l** option, but without a resource class name, the **-l** option is ignored when the command returns the list of defined resource class names.

−p *Property*
Displays attribute definitions for attributes with the specified *Property*. By default, only the definitions for public attributes are displayed. To display all attribute definitions regardless of the property, use the **-p 0** option.

Persistent Attribute Properties

0x0001	Read only
0x0002	Required for define
0x0004	Not valid for define
0x0008	Optional for define
0x0010	Selectable
0x0020	Public

Dynamic Attribute Properties

0x0020	Public

A decimal or hexadecimal value can be specified for the property. To request the attribute definitions for all attributes that have one or more properties, ″OR″ the properties of interest together and then specify the ″ORed″ value with the **-p** option. For example, to request the attribute definitions for all persistent attributes that are either required for define or optional for define, type:

```
-p 0x0a
```

−s Displays the structured data definition. Specify this option for the structured data definition to be expanded so that each element definition of the structured data attributes is displayed.

−t Specifies tabular formatted output. Each attribute is displayed in a separate column, one resource per line.

−T Writes the command's trace messages to standard error. For your software-service organization's use only.

−V Writes the command's verbose messages to standard output.

−x Suppresses header printing.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred with RMC that was based on faulty command line input.

Security

The user needs write permission for the *Resource_class* specified in **lsrsrcdef** to run **lsrsrcdef**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To list the names of all of the resource classes defined on the system, type:

   ```
   lsrsrcdef
   ```

   Output is similar to:

   ```
   class_name
   "IBM.ATMDevice"
   "IBM.Association"
   "IBM.AuditLog"
   "IBM.AuditLogTemplate"
   "IBM.Condition"
   "IBM.EthernetDevice"
   "IBM.EventResponse"
   ...
   ```

2. To list the resource class definitions for resource IBM.Host, type:

   ```
   lsrsrcdef -c IBM.Host
   ```

   Output is similar to:

   ```
   Resource Class Definition for: IBM.Host
   resource class 1:
           class_name         = "IBM.Host"
           class_id           = 8
           properties         = {"has_rsrc_insts","mtype_subdivided"}
           display_name       = ""
           description        = ""
           locator            = "NodeList"
           class_pattr_count  = 1
           class_dattr_count  = 3
           class_action_count = 0
   ```

```
        pattr_count        = 6
        dattr_count        = 47
        action_count       = 0
        error_count        = 0
        rsrc_mgr_count     = 1
rsrc_mgrs 1:
        mgr_name  = "IBM.HostRM"
        first_key = 1
        last_key  = 1
```

3. To list the resource class persistent attribute definitions for resource IBM.Host, type:

   ```
   lsrsrcdef -c -a p -p 0 IBM.Host
   ```

   Output is similar to:

   ```
   Resource Class Persistent Attribute Definitions for: IBM.Host
   attribute 1:
           program_name              = "Variety"
           display_name              = ""
           group_name                = ""
           properties                = {"read_only","invalid_for_define"}
           description               = ""
           attribute_id              = 0
           group_id                  = 255
           data_type                 = "uint32"
           variety_list              = {{1..1}}
           variety_count             = 1
           default_value             = 0
   ```

4. To list the resource persistent attribute definitions and descriptions for resource IBM.Host, type:

   ```
   lsrsrcdef -a p -p 0 -e IBM.Host
   ```

   Output is similar to:

   ```
   Resource Persistent Attribute Definitions for: IBM.Host
   attribute 1:
           program_name       = "Name"
           display_name       = "Name"
           group_name         = "General"
           properties         = {"required_for_define","public,selectable"}
           description        = "Identifies the current name of the host
         as returned by command."
           attribute_id       = 0
           group_id           = 0
           data_type          = "char_ptr"
           variety_list       = {{1..1}}
           variety_count      = 1
           default_value      = ""
   attribute 2:
           program_name       = "ResourceHandle"
           display_name       = "Resource Handle"
           group_name         = "Internal"
           properties         = {"read_only","invalid_for_define","selectable"}
           description        = "A globally unique handle that
         identifies the host. Every resource is assigned a resource
         handle, which is used internally for identifying and
         locating each resource.  The resource handle is fixed in size
         and avoids the problems of name space collisions across
         different types of resources."
           attribute_id       = 1
           group_id           = 255
           data_type          = "rsrc_handle_ptr"
           variety_list       = {{1..1}}
           variety_count      = 1
           default_value      = "0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000"
   attribute 3:
   ```

```
        program_name       = "Variety"
        display_name       = "Variety"
        group_name         = "Internal"
...
```

5. To list the public dynamic attributes for resource IBM.Host, type:

   ```
   lsrsrcdef -a d IBM.Host
   ```

   Output is similar to:

   ```
   Resource Dynamic Attribute Definitions for: IBM.Host
   attribute 1:
           program_name           = "ProcRunQueue"
           display_name           = ""
           group_name             = ""
           properties             = {"public"}
           description            = ""
           attribute_id           = 1
           group_id               = 1
           data_type              = "float64"
           variable_type          = 0
           variety_list           = {{1..1}}
           variety_count          = 1
           init_value             = 0
           min_value              = 0
           max_value              = 100
           expression      = "(ProcRunQueue - ProcRunQueue@P) >= (ProcRunQueue@P * 0.5)"
           expression_description = ""
           rearm_expression       = "ProcRunQueue < 50"
           rearm_description      = ""
           PTX_name               = ""
   attribute 2:
   ...
   ```

Files

/usr/sbin/rsct/bin/lsrsrcdef Location of the **lsrsrcdef** command.

See Also

The **lsrsrc**, **mkrsrc** commands.

The **Resource_Data_Input** file.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding RMC operations.

Author

Stephanie Beals - cluster@us.ibm.com

mkrsrc Command

Name

mkrsrc – Defines a new resource.

Synopsis

To define a new resource using data entered on the command line:

mkrsrc [**−h**] [**−v**] [**−T**] [**−V**] *Resource_class Attr=value*...

To define a new resource using data predefined in an input file:

mkrsrc [**−h**] **−f** *Resource_Data_Input_file* [**−v**] [**−T**] [**−V**] *Resource_class*

To see examples of the **mksrc** command for a Resource class:

mkrsrc [**−h**] **−e** *Resource_class*

Description

The **mkrsrc** command requests the RMC subsystem to define a new resource instance for the class specified by the *Resource_class* parameter. At least one persistent attribute name and its value must be specified either as a parameter or by a resource definition file using the **-f** option.

Prior to running **mkrsrc**, you should run the **lsrsrcdef** command to determine which attributes are required_for_define. Only attributes that have a property of required_for_define or optional_for_define can be defined using the **mkrsrc** command. The **lsrsrcdef** command also identifies the data type for each attribute. The value specified for each attribute must match this data type.

Attr=value...
Attributes of the resource being defined. When defining a new resource instance, there are specific required attributes for each resource that must be defined. These attributes can be specified as parameters on the command line or defined in an input file by using the **-f** option.

Attr The name of a persistent attribute for this resource. This attribute must have a property of required_for_define or optional_for_define. Use the **lsrsrcdef** command to check the property.

value The value for this persistent attribute. The data type for this value must match the defined data type for the value of this attribute. Use the **lsrsrcdef** command to verify the data type for each attribute.

Resource_class
The resource class name of the resource to be defined

Options

−e Displays two examples of suitable command line input for this command.
- Example of **mkrsrc** command line input for required attributes only.
- Example of **mkrsrc** command line input for both required and optional attributes.

−f *Resource_Data_Input_file*
Specifies the name of the file which contains resource attribute information.

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

−T Writes the command’s trace messages to standard error. For your software-service organization’s use only.

−v Verifies that the input file or command line is valid. For example, check that all of the attribute names specified are defined and are either required_for_define or optional_for_define.

−V Writes the command's verbose messages to standard output.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred with RMC that was based on faulty command line input.

Security

The user needs write permission for the *Resource_class* specified in **mkrsrc** to run **mkrsrc**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To create a new resource in the IBM.Host class, assuming you already know which persistent attributes are required when defining a resource of this class, type:

   ```
   mkrsrc IBM.Host Name=c175n05
   ```

2. To create a new resource in the IBM.Processor class by first generating a template to aid in the defining of these resources, type:

   ```
   lsrsrcdef -i IBM.Processor > /tmp/IBM.Processor.rdef
   ```

   Then, edit the file **/tmp/IBM.Processor.rdef** and enter values for all of the attributes, substituting the type for an appropriate value, or leaving it blank for the default value.

   Finally, type:

   ```
   mkrsrc -f /tmp/IBM.Processor.rdef IBM.Processor
   ```

3. To create two new IBM.Host resources using the information defined in file **/tmp/IBM.Host.rdef**, type:

   ```
   mkrsrc -f /tmp/IBM.Host.rdef IBM.Host
   ```

   Where the file **/tmp/IBM.Host.rdef** looks like:

   ```
   PersistentResourceAttributes::
   resource 1:
       Name        = c175n04

   resource 2:
       Name        = c175n05
   ```

4. This example creates a new resource in the IBM.Foo class. In this class, the Name and NodeList are required attributes. The Binary, SD, StringArray, and SDArray attributes are optional_for_define. This example shows how to enter the more difficult data types from the command line. The data types for the optional Binary, SD, StringArray, and SDArray attributes are self-explanatory. Type:

   ```
   mkrsrc IBM.Foo Name=c175n05 \
   NodeList={1} \
   Binary="0xaabbccddeeff00" \
   SD='[testing123,1,{2,4,6}]' \
   StringArray='{"testing 1 2 3",testing123,"testing 1 2 3"}' \
   SDArray='{["testing 1 2 3",1,{1,3,5}],[testing,2,{2,4,6}]}'
   ```

   Note: As discussed in the rmcli General Information file, attribute values for data types: structured data, array of structured data, and arrays containing strings enclosed in double quotation marks, should be enclosed in single quotation marks.

Files

/usr/sbin/rsct/bin/mkrsrc	Location of the **mkrsrc** command.

See Also

The **lsrsrc**, **lsrsrcdef**, **rmrsrc** commands.

The **Resource_Data_Input** file.

The **rmccli** General Information file.

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding RMC operations.

Author

Stephanie Beals - cluster@us.ibm.com

refrsrc Command

Name

refrsrc – Refreshes the resources within the specified resource class.

Synopsis

refrsrc [**−h**] [**−T**] [**−V**] *Resource_class*

Description

The **refrsrc** command refreshes the resources within the specified resource class. Use this command to force the Resource Monitoring and Control (RMC) subsystem to detect new instances of resources in cases where the configuration could be altered by operating system commands (**mkfs**, for example).

This command makes a request to the RMC subsystem to refresh the configuration of the resources within a resource class. The request is actually performed by the linked resource manager.

Any application that is monitoring resources in the specified resource class may receive events as the configuration is refreshed.

Resource_class	The resource class name.

Options

−h	Writes the command’s usage statement to standard output.
−T	Writes the command’s trace messages to standard error. For your software-service organization’s use only.
−V	Writes the command’s verbose messages to standard output.

Environment

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred with RMC that was based on faulty command line input.

Security

The user needs read permission for the *Resource_class* specified in **refrsrc** to run **refrsrc**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To refresh the configuration of the resources in class IBM.FileSystem, type:

   ```
   refrsrc IBM.FileSystem
   ```

Files

/usr/sbin/rsct/bin/refrsrc	Location of the **refrsrc** command.

See Also

The **lsrsrc**, **lsrsrcdef** commands.

The **rmccli** General Information file

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding RMC operations.

Author

Stephanie Beals - cluster@us.ibm.com

rmrsrc Command

Name

rmrsrc – Removes a defined resource.

Synopsis

rmrsrc [**−h**] **−s** ″*Selection_string*″ [**−T**] [**−V**] *Resource_class*

rmrsrc [**−h**] **−r** [**−T**] [**−V**] *Resource_handle*

Description

The **rmrsrc** command removes (undefines) the specified resource instance or resource instances. The **rmrsrc** command makes a request to Resource Monitoring and Control (RMC) to undefine a specific resource instance. The resource manager of the resource removes the resource.

The first format of this command requires a resource class name parameter and a selection string specified using the **-s** option. All resources in the specified resource class that match the specified selection string are removed. If the selection string identifies more than one resource to be removed, it is the same as running this command once for each resource that matches the selection string.

The second format of this command allows the actual resource handle linked with a specific resource to be specified as the parameter. It is expected that this form of the command would be more likely used from within a script.

Resource_class
The resource class name. The resource instances for this resource class that match the selection string criteria are removed.

Resource_handle
A resource handle. The resource handle must be specified using the format: ″`0x#### 0x#### 0x######## 0x######## 0x######## 0x########`″, where # is any valid hexadecimal digit. The resource handle uniquely identifies a particular resource instance that should be removed.

Options

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

−r Specifies that a resource handle is supplied as the parameter

−s ″*Selection_string*″
Specifies a selection string. All selection strings must be enclosed within either double or single quotation marks. If the selection string contains double quotation marks, enclose the entire selection string in single quotation marks. For example:

```
-s 'Name == "testing"'
```

```
-s 'Name ?= "test"'
```

Only persistent attributes may be listed in a selection string. For information on how to specify selection strings, see ″Using Expressions″ in the chapter ″Using the Monitoring Application″ of the *IBM Cluster Systems Management for Linux Administration Guide*.

−T Writes the command’s trace messages to standard error. For your software-service organization’s use only.

−V Writes the command’s verbose messages to standard output.

Environment

CT_CONTACT

When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with RMC.
2	Error occurred with CLI script.
3	Incorrect flag on command line.
4	Incorrect parameter on command line.
5	Error occurred with RMC that was based on faulty command line input.
6	No resources were found that match the selection string.

Security

The user needs write permission for the *Resource_class* specified in **rmrsrc** to run **rmrsrc**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To remove the resource with the Name c175n05 from resource class IBM.Host, type:

```
rmrsrc -s 'Name == "c175n05"' IBM.Host
```

2. To remove the resource linked with resource handle: ″0x4017 0x0001 0x00000000 0x0069684c 0x0d52332b3 0xf3f54b45″, type:

```
rmrsrc -r "0x4017 0x0001 0x00000000 0x0069684c 0x0d52332b3 0xf3f54b45"
```

Files

/usr/sbin/rsct/bin/rmrsrc	Location of the **rmrsrc** command.

See Also

The **lsrsrc**, **mkrsrc** commands.

The **rmccli** General Information file

The *IBM Cluster Systems Management for Linux Administration Guide* contains more information regarding RMC operations.

Author

Stephanie Beals - cluster@us.ibm.com

Chapter 4. CSM Scripts, Commands, and Utilities

ctsnap Command

Name

ctsnap – Gathers configuration, log, and trace information about the Reliable Scalable Cluster Technology (RSCT) components.

Synopsis

ctsnap [**-h**] [**-d** *output_directory*]

Description

The **ctsnap** command gathers configuration, log, and trace information about the RSCT components that are installed with CSM or GPFS. This command collects data only on the local node on which this command is running. Depending on the products that are installed, the following components may be included:

- Audit Log resource manager (IBM.AuditRM)
- Distributed Management Server resource manager (IBM.DMSRM)
- Event Response resource manager (IBM.ERRM)
- File System resource manager (IBM.FSRM)
- Group Services (cthags)
- Host resource manager (IBM.HostRM)
- Resource Monitoring and Control (ctrmc)
- Topology Services (cthats)
- Sensor resource manager (IBM.SensorRM)

This command is typically executed when a problem is encountered with any of these components in order to provide information to your software-service organization.

The output of **ctsnap** is a compressed tar file and a log file (**ctsnap.*host_name.nnnnnnnn*.tar.Z** and **ctsnap.*host_name.nnnnnnnn*.log**, respectively, where *nnnnnnnn* is the timestamp when the **ctsnap** command was run and *host_name* is the name of the host on which the command is running). Both files should be provided to the software-service organization. By default, these files are placed in the **/tmp/ctsupt** directory.

Error messages are written to standard error and to the **ctsnap.*host_name.nnnnnnnn*.log** file.

Options

−d *output_directory*
Identifies the output directory (The default directory is **/tmp/ctsupt**).

−h Displays the command usage information.

Exit Status

0 Command has run successfully.

1 Command was not successful.

Security

Privilege Control: Only the root user can run this command.

Examples

1. To gather RSCT support information, enter:
   ```
   ctsnap
   ```
2. To gather RSCT support information and place it in the **/tmp/mydir** directory, enter:
   ```
   ctsnap -d /tmp/mydir
   ```

Files

/usr/sbin/rsct/bin/ctsnap	Location of the **ctsnap** command.
/tmp/ctsupt	Location of the default directory that contains the output files.
tmp/ctsupt/ctsnap.*host_name.nnnnnnnn***.tar.Z**	Location of the compressed tar file that contains the collected data, where *nnnnnnnn* is a timestamp and *host_name* is the name of the host on which the command is running.
/tmp/ctsupt/ctsnap.*host_name.nnnnnnnn***.log**	Location of the log file of the command execution, where *nnnnnnnn* is a timestamp and *host_name* is the name of the host on which the command is running.

Author

Michael Schmidt - cluster@us.ibm.com

displayevent Command

Name

displayevent − Displays an event or a rearm event to a specified X-window display.

Synopsis

displayevent [**-h**] *Xdisplay*

Description

The **displayevent** command displays a message of an event or a rearm event to a particular X-window display. Event or rearm event information is captured and posted by the Event Response resource manager in environment variables that are generated by the Event Response resource manager when an event or a rearm event occurs. The **displayevent** command can be used as an action that is executed by an Event Response resource. It can also be used as a template to create other user-defined actions. See ″Event Response Resource Manager″ in the *IBM Cluster Systems Management for Linux Administration Guide* to understand how an Event Response resource runs an action command.

The following message template is displayed on the specified X-window display when an event or a rearm event for which **displayevent** is a response action occurs:

```
[ ERRM_COND_SEVERITY] [ERRM_TYPE] occurred:
  Condition: [ ERRM_COND_NAME]
  Node:  [ERRM_NODE_NAME]
  Resource: [ERRM RSRC_NAME]
  Resource Class:  [ERRM_RSRC_CLASS_NAME]
  Resource Attribute: [ERRM_ATTR_NAME]
  Attribute Type: [ERRM_DATA_TYPE]
  Attribute Value: [ERRM_VALUE]
  Time: <local time>
```

Event information is returned about the ERRM environment variables that are listed and described in ″Using Event Response Environment Variables″ in the *IBM Cluster Systems Management for Linux Administration Guide*, and also includes the following:

Local Time
: Time when the event or rearm event is observed. The actual environment variable supplied by ERRM is ERRM_TIME. This value is localized and converted to readable form before being displayed.

The **displayevent** command captures the environment variable values and uses the **gless** command to display a message to the specified X-window display.

Note: The following restriction applies. This script must be run on the host where the Event Response resource manager is running.

Options

−h Writes help information about this script to standard output. No further processing is performed.

Exit Status

0 Script has run successfully.

1 Error occurred when script was run.

Examples

1. Assume the command **/usr/sbin/rsct/bin/displayevent adminhost:0** is an action in the critical-notification response, which is associated with the ″/var space used″ condition on the resource **/var**. The threshold of the event expression defined for this condition is met, and an event occurs. The critical-notification response takes place, and **displayevent** is run. The following message is displayed on the X-window display of the machine **adminhost**:

```
Critical event occurred:
  Condition:  /var space used
  Node:  localnode
  Resource:  /var
  Resource Class:  IBM.FileSystem
  Resource Attribute: PercentTotUsed
  Attribute Type: Int32
  Attribute Value: 91
  Time: 18:43:03 03/28/01
```

Files

/usr/bin/rsct/bin/displayevent
Location of the **displayevent** command.

See Also

The ″Event Response Resource Manager″ for information about the Event Response resource manager and ″Using Event Response Environment Variables″ for information on how to use ERRM environment variables in the *IBM Cluster Systems Management for Linux Administration Guide*.

Author

Bruce Potter - cluster@us.ibm.com

logevent Command

Name

logevent − Logs event information generated by the Event Response resource manager to a specified log file.

Synopsis

logevent [**-h**] *LogFile*

Description

The **logevent** command captures event information that is posted by the Event Response resource manager in environment variables that are generated by the Event Response resource manager when an event occurs. The **logevent** command can be used as an action that is executed by an Event Response resource. It can also be used as a template to create other user-defined actions. See ″Event Response Resource Manager″ in the *IBM Cluster Systems Management for Linux Administration Guide* to understand how an Event Response resource runs an action command.

Event information is returned about the ERRM environment variables that are listed and described in ″Using Event Response Environment Variables″ in the *IBM Cluster Systems Management for Linux Administration Guide*, and also includes the following:

Local Time
Time when the event or rearm event is observed. The actual environment variable supplied by ERRM is ERRM_TIME. This value is localized and converted to readable form before being displayed.

LogFile specifies the name of the file where event information is logged. An absolute path for the *LogFile* parameter should be specified.

When *LogFile* already exists, event information is appended to it. If *LogFile* does not exist, it is created so that event information can be written to it.

Notes:

1. This command must be executed on the node where the Event Response resource manager is running.
2. The user who runs this command must have write permission for the *LogFile* where the event information is logged.
3. The size of the *LogFile* is not limited, and it will not overwrite itself. The file size will increase indefinitely unless the administrator periodically removes entries.

Options

−h Writes help information about this script to standard out. No further processing is performed.

Exit Status

0 Script has run successfully.

1 A required *logfile* is not specified.

2 The *logfile* path is invalid.

Examples

1. To log information, specify **/tmp/event.log**. The following command runs by the Event Response resource manager:

```
/usr/sbin/rsct/bin/logevent/tmp/event.log
```

Note: The **/tmp/event.log** file need not exist when the command is run.

2. To see the contents of the **/tmp/event.log** file, type:

   ```
   cat /tmp/event.log
   ```

   The following example shows a warning event for the **/var** file system (a file system resource):

   ```
   =====================================================================================
   Event reported at Mon Mar 27 16:38:03 2000

   Condition Name:                         /var space used
   Severity:                               Warning
   Event Type:                             Event
   Expression:                             PercentTotUsed>90

   Resource Name:                          /var
   Resource Class Name:                    IBM.FileSystem
   Data Type:                              CT_UINT32
   Data Value:                             91
   ```

Files

/usr/bin/rsct/bin/logevent	Location of the **logevent** command.

See Also

The ″Event Response Resource Manager″ for information about the Event Response resource manager and ″Using Event Response Environment Variables″ for information on how to use ERRM environment variables in the *IBM Cluster Systems Management for Linux Administration Guide*.

Author

Ya-Huey Juan - cluster@us.ibm.com

lsaudrec Command

Name

lsaudrec – Lists records from the audit log.

Synopsis

lsaudrec [**-h**] [**-l**] [**-n** ″*Subsystem_name*″] [**-s** ″*Selection_string*″] [**-x**] [*FieldName*...]

Description

The **lsaudrec** command lists records in the audit log. The audit log is a system-wide facility for recording information about the system’s operation. It can include information about the normal operation of the system as well as failures and other errors. It is meant to augment error log functionality by conveying the relationship of the error relative to other system activities. All detailed information about failures is still written to the error log.

Records are created in the audit log by subsystems that have been instrumented to do that. For example, the Event Response subsystem runs in the background to monitor conditions defined by the administrator and then invokes one or more actions when a condition becomes true. Because this subsystem runs in the background, it is difficult for the operator or administrator to understand the total set of events that occurred and the results of any actions that were taken in response to an event. Because the Event Response subsystem records its activity in the audit log, the administrator can view Event Response subsystem activity as well as that of other subsystems through this command.

Each record in the audit log contains named fields. Each field contains a value that provides information about the situation corresponding to the record. For example, the **Time** field indicates when the situation occurred. Each record has a set of common fields and a set of subsystem-specific fields. The common fields are present in every record in the audit log. The subsystem-specific fields vary from record to record. Their names are only significant when used with a subsystem name because they may not be unique across all subsystems. Each record is derived from a template that defines what subsystem-specific fields are present in the record and defines a format string that is used to generate a message describing the situation. The format string may use record fields as inserts. A subsystem typically has many templates.

The field names may be used as variables in a selection string to choose which records are displayed. The selection string is matched against each record by using the referenced fields of each record to perform the match. Any records that match will be displayed. The selection string is specified with the **-s** option.

A selection string is an expression composed of field names, constants and operators. The syntax of a selection string is very similar to an expression in the C programming language. For a complete description of the selection string syntax, see ″Using Expressions″ in the chapter ″Using the Monitoring Application″ of the *IBM Cluster Systems Management for Linux Administration Guide*.

Field names may also be specified as parameters to this command to choose which fields display and the order in which they display.

The common field names are:

Time
Time when the situation to which the record corresponds occurred. The value is a 64-bit integer and represents the number of microseconds since Unix Epoch (00:00:00 GMT January 1, 1970). See the constants in the following list to specify time in more user-friendly formats.

Subsystem
Identifies the subsystem that generated the record. It is a string.

Category
Identifies the category of the situation as determined by the subsystem that generated the audit log record. The category is represented as a 32-bit unsigned integer that may have the value of 0 (informational) or 1 (error).

SequenceNumber
Specifies the unique 64-bit integer that is assigned to the record. No other record in the audit log has the same sequence number.

TemplateId
Specifies the subsystem-dependent identifier that is assigned to records that have the same content and format string. This value is a 32-bit unsigned integer.

In addition to the constants in expressions that are described in the selection string reference, the following syntax for dates and times may be used with this command.

#mmddhhmmyyyy
This format consists of a sequence of decimal characters that are interpreted according to the pattern shown. The fields in the pattern are from left to right, mm=month, dd=day, hh=hour, mm=minutes, yyyy=year. For example, ″#010523042000″ corresponds to January 5, 11:04 PM, 2000. The fields may be omitted from right to left. If not present, the following are used: year defaults to the current year, minutes default to 0, hour defaults to 0, day defaults to 1, and month defaults to the current month.

#-mmddhhmmyyyy
This format is similar to the previous one but is relative to the current time and date. For example, the value #-0001 corresponds to one day ago and the value #-010001 corresponds to one month and one hour ago. Fields may be omitted, starting from the right, and are replaced by 0.

The audit records considered for display and matched against the selection string can be restricted to a specific subsystem by using the **-n** option. If this option is present, then the subsystem-specific field names may be used in the selection string as well as the common field names.

The audit records are displayed in tabular format. If any of the field names are specified as parameters, they control the fields that are displayed and the order in which they appear on each line. By default, the columns displayed are: date and time, name of the subsystem that generated the record, severity of the situation, and the subsystem-specific message that describes the situation.

FieldName identifies a field in the audit log records that is to be displayed. Multiple field names may be specified, and their order on the command line corresponds to the order in which they are displayed. If no field names are displayed, then **Time**, **Subsystem**, **Category**, and **Message** are displayed by default

Options

−h Writes help information about this script to standard out. No further processing is performed.

−l Specifies long formatted output with one entry per line. The option also displays extended fields (if applicable), which provide additional information about the recorded event. Extended fields are not displayed if **−l** is not specified.

−n ″*Subsystem_name*″
Specifies a subsystem name. If this option is present, then only records from the subsystem identified by ″*Subsystem_name*″ are considered for display. The records displayed can be further restricted by the **-s** option. If the subsystem name contains any spaces, it must be enclosed within single or double quotation marks.

−s ″*Selection_string*″
Specifies a *Selection_string* that is evaluated against each record in the audit log. All records that match the *Selection_string* are displayed.

If the *Selection_string* contains any spaces, it must be enclosed within single or double quotes. See ″Using Expressions″ in the *IBM Cluster Systems Management for Linux Administration Guide* for a complete description of the *Selection_string* syntax. The names of fields within the record may be used in the expression. If the **-n** option is not specified, then only the names of common fields may be used. See ″Description″ for a list of the field names and their data types. If the **-n** option is present, the name of any field for the specified subsystem as well as the common field names may be used.

If this option is omitted, the records that are displayed depend on the **-n** option. If the **-n** option is omitted, all records from the audit log are displayed. Otherwise, all records for the subsystem identified by the **-n** option are displayed.

−x Suppresses header printing.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with the RMC subsystem
2	Error occurred in CLI (command-line interface) program
3	Bad option on command line
4	Bad operand on command line
5	User error

Security

The user must have write permission to the **IBM.AuditLog** resource class to run **lsaudrec**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To list all records in the audit log, type:
   ```
   lsaudrec
   ```
2. To list all records that were logged in the last hour, type:
   ```
   lsaudrec -s"Time>#000001"
   ```
3. To list the time and sequence number of every record in the audit log for the subsystem **abc**, type:
   ```
   lsaudrec -s'Subsystem=="abc"' Time SequenceNumber
   ```

Files

/usr/bin/rsct/bin/lsaudrec	Location of the **lsaudrec** command.

See Also

The **rmaudrec** command.

The ″Using Expressions″ section in the *IBM Cluster Systems Management for Linux Administration Guide* provides information about selection string syntax.

Author

Gregory Laib - cluster@us.ibm.com

lssrc Command

Name

lssrc − Gets the status of a subsystem, a group of subsystems, or a subserver.

Synopsis

To get all status:

lssrc [**-h** *Host*] **−a**

To get group status:

lssrc [**-h** *Host*] **−g** *GroupName*

To get subsystem status:

lssrc [**-h** *Host*] [**−l**] **−s** *Subsystem*

To get status by PID:

lssrc [**-h** *Host*] [**−l**] **−p** *SubsystemPID*

Description

The **lssrc** command sends a request to the System Resource Controller (SRC) to get status on a subsystem, a group of subsystems, or all subsystems. The **lssrc** command sends a subsystem request packet to the daemon to be forwarded to the subsystem for a subserver status or a long subsystem status.

You can choose whether to request a short or long status for a subserver. When the **−l** flag is absent, the status request is assumed to be a short status. A short status of a subsystem, group of subsystems, or all subsystems is handled by the SRC.

When the **−l** flag is present for a subsystem, a status request is taken to the subsystem and the subsystem sends the status back. The **−l** flag is supported only for those subsystems not using signals as their communication method. For either a long or short status of a subserver, the subsystem is sent a status request packet, and the subsystem sends the status back

Error messages are written to standard error.

Options

−a Lists the current status of all defined subsystems.

−g *GroupName*
Specifies a group of subsystems to get status for. The command is unsuccessful if the GroupName variable is not contained in the subsystem object class.

−h *Host*
Specifies the foreign host on which this status action is requested. The local user must be running as root. The remote system must be configured to accept remote SRC requests. That is, the **srcmstr** daemon (see **/etc/inittab**) must be started with the **−r** flag and the **/etc/hosts.equiv** or **.rhosts** file must be configured to allow remote requests.

−l Requests that a subsystem send its current status in long form. Long status requires that a status request be sent to the subsystem; it is the responsibility of the subsystem to return the status.

−p *SubsystemPID*
Specifies a particular instance of the SubsystemPID variable to get status for, or a particular instance of the subsystem to which the status subserver request is to be taken.

−s *Subsystem*
Specifies a subsystem to get status for. The *Subsystem* variable can be the actual subsystem name or the synonym for the subsystem. The command is unsuccessful if the *Subsystem* variable is not contained in the subsystem object class.

Exit Status

0	Command has run successfully.
1	Command was not successful.

Examples

1. To get the status of all subsystems that are known on the local machine, type:
   ```
   lssrc -a
   ```
2. To get the status of a particular subsystem **cthags** subsystem, type:
   ```
   lssrc -s cthags
   ```
   The status of all instances of the **cthags** subsystem on the local machine is returned.
3. To get the status of the subsystem by PID, for example, PID 1234, type:
   ```
   lssrc -p 1234
   ```
   This gets the status of the subsystem with the subsystem PID of 1234 on the local machine.
4. To get the status of the **cthags** subsystem group, enter:
   ```
   lssrc -g cthags
   ```
   This gets the status of all instances of subsystems in the **cthags** group on the local machine.
5. To get the status of the **tester** subserver that belongs to the **srctest** subsystem, enter:
   ```
   lssrc -t tester -p 1234
   ```
   This gets the status of **tester** subserver that belongs to the **srctest** subsystem with the subsystem PID of 1234 on the local machine.

Files

/etc/objrepos/SRCsubsys	Specifies the SRC Subsystem Configuration Object Class.
/etc/objrepos/SRCsubsvr	Specifies the SRC Subserver Configuration Object Class.
/etc/objrepos/SRCnotify	Specifies the SRC Notify Configuration Object Class.
/etc/services	Defines the sockets and protocols used for Internet services.
/dev/SRC	Specifies the AF_UNIX (UNIX domain) socket file.
/dev/.SRC-unix	Specifies the location for temporary socket files.

Author

Myung Bae - cluster@us.ibm.com

msgevent Command

Name

msgevent – Sends an event or a rearm event to a specified user's console.

Synopsis

msgevent [**-h**] **user** [*tty*]

Description

The **msgevent** command displays a message of an event or a rearm event on the console of a specified user. If *tty* is specified, the message will be sent to that tty; otherwise, the **write** command chooses a tty that belongs to that user. Event or rearm event information is captured and posted by the Event Response resource manager in environment variables that are generated by the Event Response resource manager when an event or a rearm event occurs. The **msgevent** command can be used as an action that is executed by an Event Response resource. It can also be used as a template to create other user-defined actions. See the section ″Event Response Resource Manager″ in the *IBM Cluster Systems Management for Linux Administration Guide* to understand how an Event Response resource runs an action command.

The following message template is displayed on the specified console or tty when an event or a rearm event for which **msgevent** is a response action occurs:

```
[ ERRM_COND_SEVERITY] [ERRM_TYPE] occurred:
  Condition: [ ERRM_COND_NAME]
  Node:  [ERRM_NODE_NAME]
  Resource: [ERRM RSRC_NAME]
  Resource Class:  [ERRM_RSRC_CLASS_NAME]
  Resource Attribute: [ERRM_ATTR_NAME]
  Attribute Type: [ERRM_DATA_TYPE]
  Attribute Value: [ERRM_VALUE]
  Time: <local time>
```

Event information is returned about the ERRM environment variables that are listed and described in ″Using Event Response Environment Variables″ in the *IBM Cluster Systems Management for Linux Administration Guide*, and also includes the following:

Local Time
Time when the event or rearm event is observed. The actual environment variable supplied by ERRM is ERRM_TIME. This value is localized and converted to readable form before being displayed.

The **msgevent** command captures the preceding environment variable values and uses the **write** command to display a message on the specified user's console.

Note: The following restriction applies. This script must be run on the host where the Event Response resource manager is running.

Options

−h Writes help information about this script to standard out. No further processing is performed.

Exit Status

0 Script has run successfully.

1 Error occurred when script was run.

Examples

1. Assume the command **/usr/sbin/rsct/bin/msgevent root** is an action in the critical-notification response, which is associated with the ″/var space used″ condition on the resource **/var**. The threshold of the event expression defined for this condition is met, and an event occurs. The critical-notification response takes place, and **msgevent** is run. The following message is displayed on a tty belonging to root:

```
Critical event occurred:
  Condition:  /var space used
  Node:  localnode
  Resource:  /var
  Resource Class:  IBM.FileSystem
  Resource Attribute: PercentTotUsed
  Attribute Type: Int32
  Attribute Value: 91
  Time: 18:43:03 03/28/01
```

Files

/usr/bin/rsct/bin/msgevent	Location of the **msgevent** command.

See Also

The ″Event Response Resource Manager″ for information about the Event Response resource manager and ″Using Event Response Environment Variables″ for information on how to use ERRM environment variables in the *IBM Cluster Systems Management for Linux Administration Guide*.

Author

Bruce Potter - cluster@us.ibm.com

notifyevent Command

Name

notifyevent – Mails event information generated by the Event Response resource manager to a specified user ID.

Synopsis

notifyevent [**-h**] [*UserID*]

Description

The **notifyevent** command captures event information that is posted by the Event Response resource manager in environment variables that are generated by the Event Response resource manager (ERRM) when an event occurs. The **notifyevent** command can be used as an action that is executed by an Event Response resource. It can also be used as a template to create other user-defined actions.

See ″Event Response Resource Manager″ in *IBM Cluster Systems Management for Linux Administration Guide* to understand how an Event Response resource runs an action command.

Event information is returned about the ERRM environment variables that are listed and described in ″Using Event Response Environment Variables″ in the *IBM Cluster Systems Management for Linux Administration Guide*, and also includes the following:

Local Time
Time when the event or rearm event is observed. The actual environment variable supplied by ERRM is ERRM_TIME. This value is localized and converted to readable form before being displayed.

The **notifyevent** command uses the **mail** command to send event information to the specified userid. When a userid is specified, it is assumed to be valid, and it is used without verifying it. If a userid is not specified, the user who is running the command is used as the default.

Note: The following restrictions apply:

1. This command must be executed on the node where the Event Response resource manager is running.
2. The **mail** command is used to read the file.

UserID is the optional ID of the user to whom the event information will be mailed. If *UserID* is not specified, the user who is running the command is used as the default.

Options

−h Writes help information about this command to standard out. No further processing is performed.

Exit Status

0 Command has run successfully.

Examples

1. Specify **user1** to send mail to a user. The Event Response resource manager then runs the following command:

   ```
   /usr/bin/rsct/sbin/notifyevent user1
   ```

2. You can use the **mail** command to read the contents of the event information. The following example shows how a warning event for the **/var** file system (a file system resource) is formatted and logged:

```
=============================================================================
Event reported at Sun Mar 26 16:38:03 2000

Condition Name:   /var space used
Severity: Warning
Event Type:   Event
Expression:  PercentTotUsed>90

Resource Name: /var
Resource Class Name: IBM.FileSystem
Data Type: CT_UINT32
Data Value: 91
```

Files

/usr/sbin/rsct/bin/notifyevent Location of the **notifyevent** command.

See Also

The **mail** command

The ″Event Response Resource Manager″ for information about the Event Response resource manager and ″Using Event Response Environment Variables″ for information on how to use ERRM environment variables in the *IBM Cluster Systems Management for Linux Administration Guide*.

Author

Ya-Huey Juan - cluster@us.ibm.com

rmaudrec Command

Name

rmaudrec – Removes records from the audit log.

Synopsis

rmaudrec [**-h**] [**-n** ″*Subsystem_name*″] [**-s** ″*Selection_string*″] [**-V**]

Description

The **rmaudrec** command deletes records in the audit log. The audit log is a system-wide facility for recording information about the system's operation. It can include information about the normal operation of the system as well as failures and other errors. It is meant to augment error log functionality by conveying the relationship of the error relative to other system activities. All detailed information about failures is still written to the error log.

Records are created in the audit log by subsystems that have been instrumented to do that. For example, the Event Response subsystem runs in the background to monitor conditions defined by the administrator and then invokes one or more actions when a condition becomes true. Because this subsystem runs in the background, it is difficult for the operator or administrator to understand the total set of events that occurred and the results of any actions that were taken in response to an event. Because the Event Response subsystem records its activity in the audit log, the administrator can easily view Event Response subsystem activity as well as that of other subsystems. In addition, records may sometimes need to be removed explicitly, which can be done through this command.

Each record in the audit log contains named fields. Each field contains a value that provides information about the situation corresponding to the record. For example, the **Time** field indicates when the situation occurred. Each record has a set of common fields and a set of subsystem-specific fields. The common fields are present in every record in the audit log. The subsystem-specific fields vary from record to record. Their names are only significant when used with a subsystem name because they may not be unique across all subsystems. Each record is derived from a template that defines what subsystem-specific fields are present in the record and defines a format string that is used to generate a message describing the situation. The format string may use record fields as inserts. A subsystem typically has many templates.

The field names may be used as variables in a selection string to choose which records are deleted. The selection string is matched against each record by using the referenced fields of each record to perform the match. Any records that match are removed. The selection string is specified with the **-s** option.

A selection string is an expression composed of field names, constants and operators. The syntax of a selection string is very similar to an expression in the C programming language. For a complete description of the selection string syntax, see ″Using Expressions″ in the *IBM Cluster Systems Management for Linux Administration Guide*.

The common field names are:

Time Time when the situation to which the record corresponds occurred. The value is a 64-bit integer and represents the number of microseconds since Unix Epoch (00:00:00 GMT January 1, 1970). See the constants in the following list to specify time in more user-friendly formats.

Subsystem
Identifies the subsystem that generated the record. It is a string.

Category
Identifies the category of the situation as determined by the subsystem that generated the audit log record. The category is represented as a 32-bit unsigned integer that may have the value of 0 (informational) or 1 (error).

SequenceNumber
Specifies the unique 64-bit integer that is assigned to the record. No other record in the audit log has the same sequence number.

TemplateId
Specifies the subsystem-dependent identifier that is assigned to records that have the same content and format string. This value is a 32-bit unsigned integer.

In addition to the constants in expressions that are described in the selection string reference, the following syntax for dates and times may be used with this command.

#mmddhhmmyyyy
This format consists of a sequence of decimal characters that are interpreted according to the pattern shown. The fields in the pattern are from left to right, mm=month, dd=day, hh=hour, mm=minutes, yyyy=year. For example, ″#010523042000″ corresponds to January 5, 11:04 PM, 2000. The fields may be omitted from right to left. If not present, the following are used: year defaults to the current year, minutes default to 0, hour defaults to 0, day defaults to 1, and month defaults to the current month.

#-mmddhhmmyyyy
This format is similar to the previous one but is relative to the current time and date. For example, the value #-0001 corresponds to one day ago and the value #-010001 corresponds to one month and one hour ago. Fields may be omitted starting from the right and are replaced by 0.

The audit records considered for deletion and matched against the selection string can be restricted to a specific subsystem by using the **-n** option. If this option is present, then the subsystem-specific field names can be used in the selection string as well as the common field names.

It is advisable to first use the **lsaudrec** command with the same **-s** option value to list the records that are to be deleted. This minimizes the possibility of the selection string matching more records than intended.

If the **-V** option is specified and the command is completed successfully, a message that indicates the number of records that were deleted is written to standard error.

Note: The following restrictions apply:

- The command must be executed on the machine whose audit log is to have records removed.

Options

−h Writes help information about this script to standard out. No further processing is performed.

−n ″*Subsystem_name*″
Specifies a subsystem name. If this option is present, then only records from the subsystem identified by ″*Subsystem_name*″ are considered for deletion. The records deleted may be further restricted by the **-s** option. If the subsystem name contains any spaces, it must be enclosed within single or double quotation marks.

-s ″*Selection_string*″
Specifies a *Selection_string* that is evaluated against each record in the audit log. If the evaluation results in a non-zero result (TRUE), then the record is removed from the audit log. If the *Selection_string* contains any spaces, it must be enclosed within single or double quotes. See ″Using Expressions″ in the *IBM Cluster Systems Management for Linux Administration Guide* for a complete description of the selection string syntax.

The names of fields within the record may be used in the expression. If the **-n** option is not specified, then only the names of common fields may be used. See ″Description″ for a list of the common field names and their data types. If the **-n** option is present, the name of any field for the specified subsystem as well as the common field names may be used.

No records will be removed from the audit log if this option is not present.

−V Writes the verbose messages of the command to standard error.

Environment

CT_CONTACT
When the CT_CONTACT environment variable is set to a host name or IP address, the command contacts the Resource Monitoring and Control (RMC) daemon on the specified host. If the environment variable is not set, the command contacts the RMC daemon on the local system where the command is being run. The resource class or resources that are displayed or modified by the command are located on the system to which the connection is established.

Exit Status

0	Command has run successfully.
1	Error occurred with the RMC subsystem
2	Error occurred in CLI (command-line interface) program
3	Bad option on command line
4	Bad operand on command line
5	User error

Security

The user needs write permission for the **IBM.AuditLog** resource class to run **rmaudrec**. Permissions are specified in the access control list (ACL) file on the contacted system. See ″Security Considerations″ in the *IBM Cluster Systems Management for Linux Administration Guide* for details on the ACL file and how to modify it.

Examples

1. To remove all records in the audit log, type:
   ```
   rmaudrec -s"Time>0"
   ```
   or
   ```
   rmaudrec -s"SequenceNumber>=0"
   ```
2. To remove all records more than a week old, type:
   ```
   rmaudrec -s"Time <#-0007"
   ```
3. To remove all records more than a week old and created by the Abc subsystem, type:
   ```
   rmaudrec -s"Subsystem==Abc" && "Time <#-0007"
   ```

Files

/usr/bin/rsct/bin/rmaudrec	Location of the **rmaudrec** command.

See Also

The **lsaudrec** command.

The ″Using Expressions″ section in the *IBM Cluster Systems Management for Linux Administration Guide* provides information about selection string syntax.

Author

Ya-Huey Juan - cluster@us.ibm.com

rmcctrl Command

Name

rmcctrl – Manages the Resource Monitoring and Control (RMC) subsystem.

Synopsis

rmcctrl {**-a** | **-A** | **-d** | **-h** | **-k** | **-K** | **-p** | **-P** | **-s** | **-z**}

Description

The **rmcctrl** command controls the operation of the Resource Monitoring and Control (RMC) subsystem. The subsystem is under the control of the System Resource Controller (SRC) with a subsystem name of **ctrmc** and a subsystem group name of **rsct**. The RMC subsystem definition is added to the subsystem object class and then started when Reliable Scalable Cluster Technology (RSCT) is installed. In addition, an entry is made in the **/etc/inittab** file so that the RMC subsystem is automatically started at system boot.

> **Note:** While the RMC subsystem can be stopped and started by using the **stopsrc** and **startsrc** commands, it is recommended that the **rmcctrl** command be used to perform these functions.

Options

−a	Adds the RMC subsystem to the subsystem object class and places an entry at the end of the **/etc/inittab** file.
−A	Adds and starts the RMC subsystem.
−d	Deletes the RMC subsystem from the subsystem object class and removes the RMC entry from the **/etc/inittab** file.
−h	Displays the command usage message.
−k	Stops the RMC subsystem.
−K	Stops the RMC subsystem and all resource managers
−p	Enables remote client connections
−P	Disables remote client connections
−s	Starts the RMC subsystem.
−z	Stops the RMC subsystem and all resource managers, but the command does not return until the RMC subsystem and the resource managers are actually stopped.

Exit Status

0	Command has run successfully.
1	Command was not successful.

Security

Privilege Control: Only the root user should have execute (x) access to this command.

Examples

1. To add the RMC subsystem, type:
   ```
   rmcctrl -a
   ```
2. To start the RMC subsystem, type:
   ```
   rmcctrl -s
   ```

3. To stop the RMC subsystem, type:
```
rmcctrl -k
```
4. To delete the RMC subsystem, type:
```
rmcctrl -d
```

Files

/usr/sbin/rsct/bin/rmcctrl	Location of the **rmcctrl** command.

Author

Michael Schmidt - cluster@us.ibm.com

wallevent Command

Name

wallevent – Broadcasts an event or a rearm event to all users who are logged in.

Synopsis

wallevent [**-h**]

Description

The **wallevent** command broadcasts a message on an event or a rearm event to all users who are currently logged in to the host when the event or the rearm event occurs. Event or rearm event information is captured and posted by the Event Response resource manager in environment variables that are generated by the Event Response resource manager when an event or a rearm event occurs. The **wallevent** command can be used as an action that is executed by an Event Response resource. It can also be used as a template to create other user-defined actions.

See ″Event Response Resource Manager″ in the *IBM Cluster Systems Management for Linux Administration Guide* to understand how an Event Response resource runs an action command.

The following message template is displayed at the consoles of all users who are logged in when an event or a rearm event for which **wallevent** is a response action occurs:

```
Broadcast message from <user@host> <tty> at <hh:mm:ss>...
<Severity> <Event Type> occurred for the condition <Condition Name>
 on the resource <Resource Name> of <Resource Class Name> at <Event Time>
```

Event information is returned about the ERRM environment variables that are listed and described in ″Using Event Response Environment Variables″ in the *IBM Cluster Systems Management for Linux Administration Guide*, and also includes the following:

Local Time
Time when the event or rearm event is observed. The actual environment variable supplied by ERRM is ERRM_TIME. This value is localized and converted to readable form before being displayed.

The **wallevent** command captures the environment variable values and uses the **wall** command to write a message to the currently logged-in user consoles.

Note: The following restrictions apply:

1. This script must be run on the node where the Event Response resource manager is running.
2. The **wall** command is used to write a message to currently logged-in user consoles. Refer to the **wall** man page for more information on the **wall** command.

Options

−h Writes help information about this command to standard output. No further processing is performed.

Exit Status

0 Script has run successfully.

1 Error occurred when the script was run.

Examples

1. Assume the **wallevent** command is a predefined action in the critical-notification response, which is associated with the /var space used condition on the resource **/var**. The threshold of the event expression defined for this condition is met, and an event occurs. The critical-notification response takes place, and **wallevent** is run. The following message is displayed on the consoles of all logged-in users:

   ```
   Broadcast message from joe@neverland.com (pts/6) at 18:42:03...

   Critical event occurred for the Condition /var space used
   on the resource /var of JFS at 18:43:03 03/28/00
   ```

2. When a rearm event occurs for the /var space used condition on the resource **/var**, the following message is displayed on the consoles of all users who are logged in:

   ```
   Broadcast message from joe@neverland.com (pts/6) at 18:42:03...

   Critical rearm event occurred for the Condition /var space used
   on the resource /var of JFS at 18:43:03 03/28/00
   ```

Files

/usr/sbin/rsct/bin/wallevent	Location of the **wallevent** command.

See Also

The **wall** command.

The ″Event Response Resource Manager″ for information about the Event Response resource manager (ERRM) and ″Using Event Response Environment Variables″ for information on how to use ERRM environment variables in the *IBM Cluster Systems Management for Linux Administration Guide*.

Author

Ya-Huey Juan - cluster@us.ibm.com

Notices

This information was developed for products and services offered in the U.S.A.

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 terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

- IBM, AIX, AIX 5L, Netfinity, RS/6000, SP, and the e (logo) are trademarks or registered trademarks of International Business Machines Corporation.
- Equinox is a trademark of Equinox Systems, Inc.
- Linux is a registered trademark of Linus Torvalds.
- Myrinet is a trademark of Myricom, Inc.
- Red Hat and RPM are trademarks of Red Hat, Incorporated.
- Java and all Java-based trademarks and logos are registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Other company, product, or service names may be the trademarks or service marks of others.

Publicly Available Software

IBM Cluster Systems Management for Linux includes software that is publicly available:

Advanced TFTP	Advanced TFTP (atftp) is a client/server implementation of the TFTP protocol. It is licensed under the GNU General Public License (GNU GPL), which can be found at **http://www.gnu.org/copyleft/gpl.html**.
cfengine	A software package that is licensed under GPL and is used to create customization scripts.

Conserver	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.
DBD-CSV, Text-CSV_XS	Licensed by GPL or Artistic, these are dynamically loaded Perl modules.
DBI	Licensed by GPL or Artistic, this is a dynamically loaded Perl module.
fping	Copyrighted by Stanford University, this is executed as a separate binary.
Kerberos	Provides authentication of the execution of remote commands.
Perl	Practical Extraction and Report Language is licensed under the Artistic license.
Perl-to-C extensions	Practical Extraction and Report Language-to-C extensions is distributed under the Artistic license.
Pidentd	Public domain program by Peter Eriksson that implements the RFC-1413 identification server.
SQL-Statement	Licensed under GPL or Artistic, this is a dynamically loaded Perl module.
SYSLinux	SYSLinux includes PXELINUX, which CSM uses to control the behavior of network boots. SYSLinux is licensed under the GNU GPL.

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 these products 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 Cluster Systems Management 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 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 the fping open source code, see the following URL: **http://www.fping.com** (The license is imbedded in the source code.)

For the fping open source code, the following terms apply:

```
Copyright (c) 1991, 1994, 1997 Board of Trustees
            Leland Stanford Jr. University
```

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by Stanford University. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ″AS IS″ AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

For the DBD-CSF, DBI, Text-CSV_XS and SQL-Statement open source code licensed under the ″Artistic License,″ the following terms apply:

```
Definitions

"Package" refers to the collection of files distributed by the
```

```
Copyright Holder, and derivatives of that collection of files created
through textual modification.

"Standard Version" refers to such a Package if it has not been
modified, or has been modified in accordance with the wishes of the
Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights
for the package.

"You" is you, if you're thinking about copying or distributing this
Package.

"Reasonable copying fee" is whatever you can justify on the basis of
media cost, duplication charges, time of people involved, and so on.
(You will not be required to justify it to the Copyright Holder, but
only to the computing community at large as a market that must bear the
fee.)

"Freely Available" means that no fee is charged for the item itself,
though there may be fees involved in handling the
item. It also means that recipients of the item may redistribute
it under the same conditions they received it.
```

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that all copies contain the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
   a. place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
   b. use the modified Package only within your corporation or organization.
   c. rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
   d. make other distribution arrangements with the Copyright Holder.
4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
   a. distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
   b. accompany the distribution with the machine-readable source of the Package with your modifications.
   c. give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
   d. make other distribution arrangements with the Copyright Holder.
5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package’s interpreter within an executable of

yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called ″undump″ or ″unexec″ methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.
7. C subroutines (or comparably compiled subroutines in other languages supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.
8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.
9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
10. THIS PACKAGE IS PROVIDED ″AS IS″ AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

For the Advanced TFTP (atftp), cfengine, and 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

Readers' Comments — We'd Like to Hear from You

IBM Cluster Systems Management for Linux®
Technical Reference
Version 1.1

Publication No. SA22-7851-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

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.	

Readers' Comments — We'd Like to Hear from You

SA22-7851-02

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie NY 12601-5400

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Program Number: 5765–E88

SA22-7851-02