IBM Cluster Systems Management for Linux®

Software Planning and Installation Guide

Version 1.2

SA22-7853-03

IBM Cluster Systems Management for Linux®

Software Planning and Installation Guide

Version 1.2

SA22-7853-03

Note!

Before using this information and the product it supports, read the information in “Notices” on page 59.

Fourth Edition (July 2002)

This edition applies to version 1, release 2 of IBM Cluster Systems Management (CSM) licensed program (product number 5765–E88) and to all subsequent releases and modifications of this product until otherwise indicated in new editions. This edition replaces SA22–7853–02. Significant changes or additions to the text and illustrations are indicated by a vertical line (|) to the left of the change.

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

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

FAX (United States & Canada): 1+845+432-9405
FAX (Other Countries):
Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)
IBM Mail Exchange: USIB6TC9 at IBMMAIL
Internet e-mail: 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 provides planning, installation, backup and recovery, and troubleshooting information related to getting the Cluster Systems Management for Linux product up and running. The information includes the specified operating environment for the Cluster Systems Management (CSM) suite of tools and describes how to install and set up a CSM cluster on an existing group of nodes or as a full installation of Linux and CSM on the nodes of the cluster.

Who should use this book

This guide is intended for system administrators who want to use IBM Cluster Systems Management. It describes tools that are provided to make the installation of Cluster Systems Management easier. The system administrator should have experience in UNIX® administration and networked systems.

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:

- IBM CSM for Linux: Hardware Planning and Control Guide, SA22–7856-03
- IBM CSM for Linux: Software Planning and Installation Guide, SA22–7853–03
- IBM CSM for Linux: Adminstration Guide, SA22–7873–02

The following references contain information about IBM Reliable Scalable Cluster Technology (RSCT) for Linux:

- IBM RSCT for Linux: Guide and Reference, SA22–7892–00
- IBM RSCT for Linux: Technical Reference, SA22–7893–00
- IBM RSCT for Linux: Messages, GA22–7894–00
- IBM RSCT for Linux: Group Services Programming Guide and Reference, SA22–7888–00

How to obtain publications

The IBM Cluster Systems Management (CSM) 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 information is available on the CD-ROM in the root directory (**/**).

The file names are as follows:

- *IBM CSM for Linux: Software Planning and Installation Guide* (am7lxins.pdf)
- *IBM CSM for Linux: Administration Guide* (am7lxadm.pdf)
- *IBM CSM for Linux: Hardware Planning and Installation Guide* (am7lxhwc.pdf)

These IBM Cluster Systems Management (CSM) for Linux publications were also available at the time of this release at http://www.ibm.com/servers/eserver/clusters/library.

How to contact IBM

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

Chapter 1. Software Planning for Cluster Systems Management

The planning and installation process described in this book helps a system administrator get IBM Cluster Systems Management for Linux, hereafter known as Cluster Systems Management (CSM), up and running easily. The CSM application is available for Linux as an installation image in a directory or on the CSM CD-ROM. You can install CSM by setting up a management server and managed nodes on existing Linux systems or by doing a full installation of the operating system and CSM on the nodes of the cluster (not the management server, which must be set up first).

Note that this book assumes that your hardware has already been set up. If you have not done so already, see the *CSM for Linux: Hardware Planning and Control Guide* for information on the hardware requirements and set up tasks.

Understanding CSM software requirements

IBM Cluster Systems Management has requirements for non-IBM software as well as IBM-developed software. As a convenience, the required non-IBM software, that is not part of the Red Hat® distribution, is included on the CSM CD-ROM. Unless otherwise specified, the software is required on the management server and on the managed nodes.

CSM packages

The following packages comprise the IBM Cluster Systems Management for Linux product:

1. csm.client 1.2 (required on the managed nodes only)
2. csm.core 1.2
3. csm.server 1.2 (required on the management server only)
4. csm.dsh 1.2 (required on the management server only)
5. csm.gui.dcem 1.2 (installed on the management server)
6. rsct.core 2.2.1.10
7. rsct.core.utils 2.2.1.10
8. src 1.1.1.0

Required IBM non-CSM software

The following non-CSM IBM software is required, and is provided by IBM:

1. IBMJava2–JRE 1.3.0

Required non-IBM software

The Red Hat Linux 7.1 or 7.2 distribution is required.

The Red Hat 7.1 distribution includes the following required non-IBM software:

1. expect 5.31 (required on management server only)
2. tcl 8.3.1 (required on management server only)
3. tk 8.3.1 (required on management server only)
4. XFree86–libs 4.0.3 (required on management server only)
5. dhcp 2.0 (required on management server only)
6. freetype 2.0.3
7. glibc 2.2.2
8. libstdc++ 2.96

9. pdksh 5.2.14
10. perl 5.6.0
11. nfs-utils 0.3.1 (required on management server only)
12. rdist 6.1.5

The Red Hat 7.2 distribution includes the following required non-IBM software:

1. expect 5.32 (required on management server only)
2. tcl 8.3.1 (required on management server only)
3. tk 8.3.1 (required on management server only)
4. XFree86–libs 4.1.0 (required on management server only)
5. dhcp 2.0 (required on management server only)
6. freetype 2.0.3
7. glibc 2.2.4
8. libstdc++ 2.9.6
9. pdksh 5.2.14
10. perl 5.6.0
11. nfs-utils 0.3.1 (required on management server only)
12. rdist 6.1.5

Required non-IBM software provided by IBM

IBM provides the following required software:

1. atftp 0.3
2. syslinux 1.53

Compatibility with General Parallel File System

It is mandatory that the CSM client RPM be installed either on all of the General Parallel File System for Linux (GPFS) nodes or on none of them. The Resource Monitoring and Control (RMC) subsystem requires that the definitions of its resource classes be the same on all nodes within the GPFS cluster. If the CSM client RPM is installed on only some of the GPFS nodes, some of the RMC daemons will not be able to join their peer group successfully, an undesirable situation. The CSM management server node should not be part of the GPFS cluster because it adds other resource classes to RMC, and it is not practical or desirable to install the CSM server RPM on all of the GPFS nodes.

GPFS and CSM should be using the same level of RSCT code. The current release of RSCT is Version 2.2.1.0.

Understanding CSM installation

A CSM cluster consists of one node that acts as the management server, as well as other cluster nodes that are clients of the management server. All of these nodes must have both the operating system and the CSM software installed on them. For the management server, you will always install CSM only. For the nodes, you have a choice. You can either install CSM alone, or with the Linux operating system. For a *CSM-only* installation, it is assumed that the Red Hat Linux operating system has already been installed on the nodes. A *full* installation involves installing Red Hat Linux, using Red Hat Kickstart, along with CSM on both the nodes and the management server.

Estimating the time needed for node installation

The estimated length of time for installing CSM only on a node is a minimum of 2 minutes. The estimated length of time for doing a full install (installing both CSM and Linux on a node) is a minimum of 15 minutes. However, depending on the network traffic on your system, the number of packages you are installing, and the number of nodes you are installing simultaneously, the length of time it takes to install your system can vary greatly.

Overview of management server installation

The process of installing the management server involves basic tasks like establishing name service on the management server, and installing the CSM commands accessible on the management server. You will also need to install CSM on the management server, which you will do via the **installms** command. The **installms** command establishes the necessary CSM directories and files, and installs CSM, as well as the Open Source and Red Hat software that is required.

Management server installation also involves configuring the remote shell, for issuing commands from the management server to the nodes. Although rsh is the default remote shell, CSM also supports **secure remote command**. The DSH_REMOTE_CMD environment variable is provided by CSM to allow you to select which remote shell you wish to use.

Overview of CSM-only installation

A CSM-only installation refers to installing CSM on the nodes of your cluster alone (without the operating system). Among other tasks, a CSM-only installation involves defining the nodes of the cluster and then installing CSM on each node. You will use the **definenode** command to define your nodes, but before you can do this, you need to record the hardware and software attributes of each node on a set of node attribute templates. You will then refer to these templates when you actually define the nodes. Depending on the complexity of your system, you can put the information from the node attribute templates into a file (called a nodedef file), which you can supply to the **definenode** command. If your system is not so complex, you can simply enter the attributes on the command line with the **definenode** command.

Finally, to install CSM on the nodes of the cluster, you will use the **updatenode** command.

Overview of full installation

A full installation of CSM involves installing both CSM and the operating system on the nodes of the cluster. You will use the **definenode** command to define your nodes, but before you can do this, you need to record the hardware and software attributes of each node on a set of node attribute templates. You will then refer to these templates when you actually define the nodes. Depending on the complexity of your system, you can put the information from the node attribute templates into a file (called a nodedef file), which you can supply to the **definenode** command. If your system is not so complex, you can simply enter the attributes on the command line with the **definenode** command.

Next, you will need to create a Linux Kickstart configuration file to specify node configuration data. A Kickstart node configuration template is provided, which you can modify, if you wish. After you have prepared your Kickstart configuration template, you will use the **csmsetupks** command, in conjunction with the Kickstart configuration file, to generate a configuration file for each of the nodes in the cluster.

The last portion of the full install process involves installing the software on the nodes. You will use the **installnode** command to install both CSM and the operating system on the nodes.

Proceeding with the installation

The CSM installation instructions are contained in three separate chapters; one for installing the management server, another for installing CSM only on the nodes, and yet another for installing both CSM and the operating system on the nodes. There is also a chapter that contains information for migrating from one level of CSM to another. If you are migrating your system, you should start with that chapter.

You should read the procedures in these chapters carefully, and be familiar with them, before beginning the installation. It is also a good idea to have all the books in the CSM library close at hand, because the instructions ask you to refer to them from time to time. You also may want to know more about the commands used during installation, so the *CSM for Linux: Administration Guide* is essential. See “Related information” on page vii for a list of the other books in the CSM library.

Depending on the type of installation you want to perform, go to one of the following sections:

- Chapter 6, “Migrating to the latest level of CSM” on page 27.
- Chapter 2, “Installing the Management Server” on page 5.
- Chapter 3, “Installing only CSM on the cluster nodes” on page 9.
- Chapter 4, “Installing CSM and the operating system on the nodes” on page 15.

Chapter 2. Installing the Management Server

This chapter provides step-by-step instructions for installing CSM on the management server. Before continuing, ensure that you are running either the Linux Red Hat 7.1 or 7.2 operating system on the management server.

You should read these procedures carefully and be familiar with them before beginning the installation and set up tasks. It is also a good idea to have the *CSM for Linux: Administration Guide* close at hand in case you need to know more details about the commands used in the installation process.

Step 1. Register the host names of the nodes being defined

Register the host names of the nodes that are being defined to the cluster with the nameserver or add them to the **/etc/hosts** file on the management server.

Step 2. Create the /csminstall partition

For a full installation, IBM suggests that you create a Linux partition called **/csminstall** that consists of 1.5 GB of space for a full installation, which includes the operating system, or 100MB for a CSM-only installation. Note that for full installation, the 1.5 GB of space required is per Linux operating system — if you have both Red Hat Linux 7.1 and 7.2, double this amount. The **/csminstall partition** will hold the required RPM packages for installation.

For a CSM-only installation, it is not necessary to create a separate partition. You can simply put the CSM software and files in the root partition.

If you are using Red Hat Linux 7.2, you can use fdisk to create the partition. If you are using Red Hat Linux 7.1, you can use either fdisk or cfdisk to create the partition.

Step 3. Update the $PATH and $MANPATH variables

Add **/opt/csm/bin** to your $PATH variable, and, to access the CSM man pages, add **/opt/csm/man** to your $MANPATH variable on the management server.

Step 4. Install the csm.core file set

1. Mount the CSM CD ROM as follows:
   ```
   mount /dev/cdrom /mnt/cdrom
   ```
2. Install the csm.core file set as follows:
   ```
   rpm -i /mnt/cdrom/csm.core-*
   ```

Note: It is important that you do not unmount the CD-ROM once it has been mounted.

Step 5. Install the CSM software

The **installms** command performs the tasks that are necessary to make this system a management server and installs the required Open Source and Red Hat software. It installs the Open Source, Red Hat, and IBM software listed in “Understanding CSM software requirements” on page 1 on the management server automatically, if it is not already installed or if it is installed at a previous level. Note that you can run the **installms** command as many times as you wish (if the management has already been set up, you will not get an error).

IBM suggests that you set up the **/csminstall** partition before you run **installms**. You should have done this in “Step 2. Create the /csminstall partition”.

Note that the CSM CD ROM also needs to be mounted before using the **installms** command. You should have already done this in "Step 4. Install the csm.core file set" on page 5.

1. Log in as root to the machine that is to become the management server.
2. Run the **installms** command with the **–p** (pkg_path) flag, followed by the mount point of the CD-ROM.

   ```
   installms -p /mnt/cdrom
   ```

   a. Next, the program asks you to insert the Linux distribution CD-ROMs in the drive. The program automatically mounts and unmounts the CD-ROMs, as needed, and copies the Red Hat RPMs that are required to install CSM.

   b. The CSM, RPM, and Open Source packages are used to install the code and initialize the server. The **installms** command determines whether the code needs to be installed (if it is missing or back level) and then installs or replaces the required packages, as necessary. The **installms** command is located in **/opt/csm/bin**.

The **installms** command establishes the CSM directories and the required files when it is run from the management server. For more information about these files and directories, see Appendix B, "Understanding the installation directories and files" on page 35.

For more information about the **installms** command, see the man page or the *CSM for Linux: Administration Guide*.

Step 6. Deciding which remote shell to use

A distributed shell program (**dsh**) is used to issue remote commands from the management server to the nodes. It is contained in the **csm.dsh** RPM and installed by the **installms** command. The **dsh** program uses a remote shell of your choice to issue remote commands to the managed nodes.

The default shell is **rsh**, but you can specify a different remote shell with the DSH_REMOTE_CMD environment variable. If you wanted to switch from the default **rsh** shell and use the **secure remote command** instead, you would specify the DSH_REMOTE_CMD environment variable, with the full path name of the remote shell command. For example:

```
DSH_REMOTE_CMD=/usr/bin/ssh
```

Step 7. Verify the installation

To verify that the management server was installed correctly, issue the **lscondition**, **lsresponse**, or **nodegrp** command, as shown below.

Issue the **lscondition** command as follows:

```
export CT_MANAGEMENT_SCOPE=1
lscondition
```

If the management server is installed correctly, you will see output similar to the following:

```
Displaying condition information:

Name                                 MonitorStatus
"AnyNodePagingPercentSpaceFree"      "Not monitored"
"NodeReachability"                   "Not monitored"
"CFMRootModTimeChanged"              "Not monitored"
"AnyNodeFileSystemInodesUsed"        "Not monitored"
"AnyNodeVarSpaceUsed"                "Not monitored"
"AnyNodeProcessorsIdleTime"          "Not monitored"
"AnyNodeFileSystemSpaceUsed"         "Not monitored"
"UpdatenodeFailedStatusChange"       "Not monitored"
"AnyNodeTmpSpaceUsed"                "Not monitored"
"NodeGroupMembershipChanged"         "Not monitored"
"NodeChanged"                        "Not monitored"
"NodePowerStatus"                    "Not monitored"
```

Issue the **lsresponse** command as follows:

```
export CT_MANAGEMENT_SCOPE=1
lsresponse
```

If the management server is installed correctly, you will see output similar to the following:

```
Displaying response information:

ResponseName
"LogEventsAnyTime"
"UpdatenodeFailedStatusResponse"
"CFMModResp"
"EmailEventsToRootAnyTime"
"BroadcastEventsAnyTime"
"LogOnlyToAuditLogAnyTime"
"CFMNodeGroupResp"
"MsgEventsToRootAnyTime"
"DisplayEventsAnyTime"
```

Issue the **nodegrp** command as follows:

```
export CT_MANAGEMENT_SCOPE=1
nodegrp
```

If the management server is installed correctly, you will see output similar to the following:

```
NetfinityNodes
RedHat71Nodes
KickstartNodes
PreManagedNodes
RedHat72Nodes
RedHatNodes
AllNodes
LinuxNode
ManagedNodes
```

Chapter 3. Installing only CSM on the cluster nodes

This chapter tells you how to install only CSM on the nodes of the cluster. It assumes that the operating system has already been installed on the nodes, and that the operating system and CSM have already been installed on the management server.

Step 1. Configure the remote shell

A distributed shell program (**dsh**) is used to run commands on the nodes. It is contained in the **csm.dsh** RPM and installed by the **installms** command. The **dsh** program uses the remote shell of your choice to issue remote commands to the managed nodes from the management server (**rsh** is the default, but “Step 6. Deciding which remote shell to use” on page 6 showed you how choose another shell). To enable the remote shell, you must perform the following tasks on **each node** before **dsh** is installed. Note that these tasks are only necessary when doing a CSM-only installation.

- If you are using **rsh**, make sure the rsh-server RPM is installed and running. If you are using another remote shell, make sure its daemon is installed and running.
- Security must be set up on each node in such a way that **dsh** is allowed to run commands on that node. If you are using **rsh**, add the management server host name to the **/root/.rhosts** file on the nodes that will be managed nodes. If you are using another shell, you must fulfill the requirements for installing and using that shell.

Note: Be aware that the **dsh** command does not provide the set up for a specific security configuration. The user is responsible for fulfilling the particular security obligations of a specified security environment. At a minimum, you can configure **rsh** with the **/root/.rhosts** file on nodes. A more secure environment might have Kerberos configured or might be using some type of shell that conforms to the IETF (Internet Engineering Task Force) secure remote command protocol.

To ensure that **dsh** will work on each of the nodes, use the remote shell to run a remote command from the node that will be the management server to each node that will be in the cluster. For example you can run the following command for each node:

```
dsh -n nodename date
```

In the example above, if **dsh** is working, the **date** command returns a list of the nodes in the cluster, showing the date for each node.

Note: The Distributed Command Execution Environment (DCEM) graphical user interface is provided to make it easy to use the **dsh** command functions. See the **dcem** man page and the *CSM for Linux: Administration Guide* for information on how to start up and use DCEM.

For more information on **dsh**, see the man page or the commands chapter of *CSM for Linux: Administration Guide*.

Step 2. Configure hardware control points and remote console

If you have not done so already, you need to configure the CSM hardware control functions; remote power and remote console. Refer to *CSM for Linux: Hardware Planning and Control Guide* and follow the instructions provided there. When you have completed the remote power and remote console configuration, you are ready to go on to the next step.

For remote power, make sure the **systemid** command has been run.

Step 3. Complete the node planning templates

This step involves gathering and recording information about the nodes in your cluster. You enter the hardware-related information about your nodes on the *Hardware Node Attribute Template* and you enter the software-related information about your nodes on the *Software Node Attribute Template*.

To complete the node attribute templates, do the following:

1. Go to the *CSM for Linux: Hardware Planning and Control Guide* for instructions on filling out the Hardware Node Attribute Template. Note that the *CSM for Linux: Hardware Planning and Control Guide* contains a completed example of the template. The blank template is provided in this book in Appendix A, “Node attributes templates” on page 31. When you have completed the Hardware Node Attribute Template, return to this step.
2. Go to Appendix A, “Node attributes templates” on page 31 and follow the instructions for completing the Software Node Attribute Template. After you have done this, you will be ready to go on to “Step 4. Create a nodedef file (optional)”.

Step 4. Create a nodedef file (optional)

In the last step, you completed the node planning templates for your nodes. When defining your nodes (in an upcoming step) the information contained in the completed node planning templates needs to be supplied to the **definenode** command. It can either be entered into a node definition (**nodedef**) file, or it can be entered at the command line. A **nodedef** file allows you to enter the host names of the nodes, and all the attributes that you wish to define in a file and thus avoid the error-prone task of having to type all of the information on the command line. If you intend to use a **nodedef** file, start with the sample file in **/opt/csm/install/nodedef.sample** and complete the information from the node planning template that you completed earlier. For large numbers of nodes, IBM suggests using the **nodedef** file and not the command line. For a small number of nodes, you can simply enter the information on the management server command line with the **definenode** command (in the next step).

The following is an example of a typical **nodedef** file.

```
# Cluster System Management Node Definition File
#
# Node Definition File Rules
#
# 1.  A stanza header consists of the node name followed by a colon (:).
#
# 2.  Attribute lines must take the form of Attribute=Value.
#
# 3.  If a line is not blank, a comment, a header, or an attribute=value
#     pair then the command will skip to the next valid header.
#
# 4.  A warning message will be issued for invalid attributes but the
#     code will continue.
#
# 5.  Each line of the file may have only one header or attribute definition.
#
# 6.  Only one stanza may exist for each node.
#
# 7.  If the header keyword is "default" then the attribute values in
#     the stanza are considered default values for subsequent node definitions.
#
# 8.  Default stanzas may be specified multiple times and at any point
#     in a definition file. The values apply to all definitions following
#     it in the file. The default values are cumulative. That is, a default
#     attribute value will stay set until it is explicitly unset or changed.
#
# 9.  To "turn off" a default value set the attribute to nothing (blank).
#
# 10.  When a specific value for an attribute is provided in the node
#     stanza it will take priority over any default value that had been set.
```

```
#
# 11. Comments beginning with the pound sign (#) may be added to the
#     file. Comments must be on a separate line.
#
# 12. When parsing the file, tab characters and spaces are ignored.
#
# 13. Some attribute values will be set by default by the definenode
#     command when this file is used as input.
#
#     The following attributes will be set based on what is installed
#     on the management server.
#
#     InstallOSName - "Linux"
#     InstallDistributionName - Linux distribution  (ex. RedHat)
#     InstallDistributionVersion - Linux distribution version
#     InstallPkgArchitecture - (ex. "i386")
#     InstallCSMVersion - version of CSM
#
#
# define a set of netfinity nodes
#
#
# nfnode01:
#     ManagementServer=c5bs.ppd.xyz.com
#     HWControlPoint=asm04
#     PowerMethod=netfinity
#     ConsoleMethod=esp
#     ConsoleServer=esp15
#     ConsoleServerNumber=1
#     ConsolePortNumber=0
#     InstallOSName=Linux
#     InstallCSMVersion=1.3.0
#     InstallDistributionVersion=7.2
# nfnode02:
#     ManagementServer=c5bs.ppd.xyz.com
#     HWControlPoint=asm04
#     PowerMethod=netfinity
#     ConsoleMethod=esp
#     ConsoleServer=esp15
#     ConsoleServerNumber=2
#     ConsolePortNumber=0
#     InstallOSName=Linux
#     InstallCSMVersion=1.3.0
#     InstallDistributionVersion=7.1
# nfnode03:
#     ManagementServer=c5bs.ppd.xyz.com
#     HWControlPoint=asmn02
#     PowerMethod=netfinity
#     ConsoleMethod=esp
#     ConsoleServer=esp16
#     ConsoleServerNumber=3
#     ConsolePortNumber=0
#     InstallOSName=Linux
#     InstallCSMVersion=1.3.0
#     InstallDistributionVersion=7.1
# nfnode10:
#     ManagementServer=c5bs.ppd.xyz.com
#     InstallOSName=Linux
#     PowerMethod=netfinity
#     ConsoleMethod=esp
#     HWControlPoint=asm02
#     ConsoleServer=esp15
#     ConsoleServerNumber=2
#     ConsolePortNumber=0
#     InstallCSMVersion=1.2.0
#     InstallDistributionVersion=7.2
#
```

```
#
# using default values.
#
# default:
#     PowerMethod=netfinity
#     ConsoleMethod=esp
#     ConsoleServer=esp15
#     ConsoleServerNumber=2
#     ConsolePortNumber=0
#     InstallOSName=Linux
#     InstallCSMVersion=1.2.0
#     HWControlPoint=asm03
#     ManagementServer=c5bs.ppd.xyz.com
#     InstallDistributionVersion=7.2
# nfnode11:
#     ConsolePortNumber=1
# nfnode12:
#     HWControlPoint=asm04
#     ConsolePortNumber=2
# nfnode13:
#     HWControlPoint=asm05
#     ConsolePortNumber=3
```

If you don’t intend to use a **nodedef** file, skip ahead to the next step.

See the **nodedef** man page or *CSM for Linux: Administration Guide* for more details about the node definition file.

Step 5. Define the nodes of the cluster

After the management server is installed, (via the **installms** command), you need to define all of the nodes in the cluster. To define the nodes of the cluster, you use the **definenode** command on the management server node. Before running the **definenode** command, you should have:

- Gathered information about your nodes and entered it into the node planning templates. If you have not done this yet, see “Step 3. Complete the node planning templates” on page 15.
- Decided how you were going to supply the node definitions with the **definenode** command (via a **nodedef** file or from the command line).
- Created the **nodedef** file, if you don’t want to enter the node definitions from the command line. If you haven’t done this yet, but need to, see “Step 4. Create a nodedef file (optional)” on page 10.

Note that the **definenode** command simply creates definitions for the nodes in the cluster. It does not actually install the nodes. Node installation is done later with the **updatenode** command.

Using a nodedef file to specify nodes

To define nodes with a **nodedef** file, issue the **definenode** command, on the management server, and specify the full path name of the **nodedef** file. For example:

```
definenode -f /tmp/nodedef
```

Using the command line to specify nodes

To define a node from the command line, without hardware control and with all the default attributes, issue the **definenode** command, on the management server, and specify the host name:

```
definenode -n hostname
```

For example, to define a node with a host name of **clsn02.ppd.pok.ibm.com**, you would type:

```
definenode -n clsn02.ppd.pok.ibm.com
```

To define a node's hardware attributes from the command line, issue the **definenode** command, followed by the attributes you want to assign. The sample of a completed Hardware Node Attributes Template, from the *CSM for Linux: Hardware Planning and Control Guide*, was used to create the example below.

Setting a node's *HWControlNodeID* attribute to the short host name of the node can simplify the node definition process. The following **definenode** command example defines the node short host names as the *HWControlNodeID* attribute values. If the *HWControlNodeID* attribute values were not set to the node short host names, then the **nodedef** file would be used to specify each attribute value. If a node's *HWControlNodeID* is the short host name, then the following command can be run once to define all nodes attached to the hardware control point:

```
definenode -n clsn01.pok.ibm.com -c 20 -H mgtn03.pok.ibm.com:10,- mgtn04.pok.ibm.com:10 \
-C mgtn02.pok.ibm.com:1:0:16,mgtn05.pok.ibm.com:2:0:16 PowerMethod=netfinity ConsoleMethod=esp
```

All *HWControlNodeID* node attribute values attached to a hardware control point must be unique. For Netfinity xSeries 330 and 342 nodes, the *HWControlNodeID* value must match the Netfinity text ID set in the hardware. If the *HWControlNodeID* values are changed to the short host names of the nodes, then the **systemid** command must be subsequently run to correctly set the new user ID and password information in the CSM database.

See the man page or *CSM for Linux: Administration Guide* for details on **definenode** command line syntax and more examples of using the command.

Step 6. Verify the node definitions

After **definenode** has run, the management server has been set up with all the node information for CSM, and you are now ready to verify the node definitions. This section tells you how to verify and customize the cluster node definitions before the nodes are installed. Since the actual node installation has not happened yet, you can make changes to any of the node definitions at this time.

To determine whether the nodes have been defined, issue the **lsnode** command from the management server:

```
lsnode
```

The system responds with a line for each node that was successfully defined. If a node has not been defined, it will not appear in the output for **lsnode**.

To display all the information about each node, use the **lsnode** command, from the management server, with the –l (lowercase l, not uppercase i) option:

```
lsnode -l
```

The system responds with a list (output), containing extended information, for each node that was successfully defined. If a node has not been defined, it will not appear in the output for **lsnode**. Note that some of the attributes for a node might have null values at this point.

Note: If a node has not been defined, it will not be installed.

If something needs to be corrected, you can either remove the node that was not successfully defined and then rerun the **definenode** command with the correct arguments, or you can make the necessary changes to the attributes of the node.

To remove a node before redefining it, use the **rmnode** command on the management server:

```
rmnode hostname
```

To change the attributes of a node, use the **chnode** command from the management server:

```
chnode hostname attr=value
```

See the man pages or the *CSM for Linux: Administration Guide* for more information about the **lsnode**, **chnode**, and **rmnode** commands.

Step 7. Install CSM on the nodes

The **updatenode** command installs and updates the CSM software and files on the nodes of the cluster. The nodes on which the **updatenode** command is run must already have the operating system installed.

Before the **updatenode** command can be run, ensure that the following tasks have been completed on the management server:

- The **dsh** command must be available to perform remote commands on the nodes. In other words, security must be set up on each node in such a way that **dsh** is allowed to run commands on that node. **dsh** is used to perform an NFS mount of **/csminstall** to each node. .
- The **installms** and **definenode** commands must have been run successfully.

Nodes can be in one of three *Modes* at any given time during installation. When a node has only been defined, but not actually added to the cluster, it is in **PreManaged** mode. After a node has been defined and added to the cluster with the **updatenode** command, its Mode is **Managed**.

The **updatenode** command is run on nodes that have already been defined to the system. When **updatenode** is run, each node that was just defined (whose *Mode* attribute is **PreManaged**) becomes a managed node of the cluster. Note that if the **updatenode** command is not successful, the node’s Mode attribute remains **PreManaged**.

/var/log/csm/updatenode.log and **/var/log/csm/install.log** on each node, can be reviewed to verify that the node was installed successfully.

To install the premanaged nodes of your cluster, type:

```
updatenode -P
```

By default, **updatenode** installs CSM and distributes configuration files to the nodes. However, you can designate that CSM perform only a subset of the default tasks via command line options. If you want to do this, see the man page or the *CSM for Linux: Administration Guide* for more information.

Step 8. Verify the installation

To verify the CSM-only installation, use the **lsnode** command, with the –a flag, from the management server:

```
lsnode -a Mode
```

lsnode returns a list of the active nodes in the cluster, along with their *Mode* attribute. Nodes that have been successfully installed are set to **Managed**.

Chapter 4. Installing CSM and the operating system on the nodes

This chapter tells you how to install both CSM and the operating system on the nodes of the cluster. It assumes that the operating system and CSM have already been installed on the management server.

Step 1. Set the boot order for the nodes

You must set the boot order for each node in your cluster, and you must set it in the BIOS (Basic Input/Output System). During booting, press the F1/delete key when you are prompted. The boot order should be set as follows:

1. Diskette
2. CD-ROM
3. Network
4. Hard disk

Step 2. Configure hardware control points and remote console

If you have not done so already, you need to configure the CSM hardware control functions; remote power and remote console. Refer to *CSM for Linux: Hardware Planning and Control Guide* and follow the instructions provided there. When you have completed the remote power and remote console configuration, you are ready to go on to the next step.

For remote power, make sure the **systemid** command has been run.

Step 3. Complete the node planning templates

This step involves gathering and recording information about the nodes in your cluster. You enter the hardware-related information about your nodes on the *Hardware Node Attribute Template* and you enter the software-related information about your nodes on the *Software Node Attribute Template*.

To complete the node attribute templates, do the following:

1. Go to the *CSM for Linux: Hardware Planning and Control Guide* for instructions on filling out the Hardware Node Attribute Template. Note that the *CSM for Linux: Hardware Planning and Control Guide* contains a completed example of the template. The blank template is provided in this book in Appendix A, “Node attributes templates” on page 31. When you have completed the Hardware Node Attribute Template, return to this step.
2. Go to Appendix A, “Node attributes templates” on page 31 and follow the instructions for completing the Software Node Attribute Template. After you have done this, you will be ready to go on to “Step 4. Create a nodedef file (optional)”.

Step 4. Create a nodedef file (optional)

In a previous step, you completed the node planning templates for your nodes. When defining your nodes (in an upcoming step) the information contained in the completed node planning templates needs to be supplied to the **definenode** command. It can either be entered into a node definition (**nodedef**) file, or it can be entered at the command line. A **nodedef** file allows you to enter the host names of the nodes, and all the attributes that you wish to define in a file, and thus avoid the error-prone task of having to type all of the information on the command line. If you intend to use a **nodedef** file, start with the sample file in **/opt/csm/install/nodedef.sample** and complete the information from the node planning template that you completed earlier. For large numbers of nodes, IBM suggests using the **nodedef** file and not the command line. For a small number of nodes, you can simply enter the information on the command line with the **definenode** command.

See "Step 4. Create a nodedef file (optional)" on page 10 for an example of a typical **nodedef** file.

If you do not intend to use a **nodedef** file, skip ahead to the next step.

See the **nodedef** man page or *CSM for Linux: Administration Guide* for more details about the node definition file.

Step 5. Define the nodes of the cluster

After the management server is installed, (via the **installms** command), you need to define all of the nodes in the cluster. To define the nodes of the cluster, you use the **definenode** command. Before running the definenode command, you should have:

- Gathered information about your nodes and entered it into the node planning templates.
- Decided how you were going to supply the node definitions with the **definenode** command (via a **nodedef** file or from the command line).
- Created the **nodedef** file, if you do not want to enter the node definitions from the command line.

Note that the **definenode** command simply creates definitions for the nodes in the cluster. It does not actually install the nodes. Node installation is done later with the **installnode** command.

Using a nodedef file to specify nodes

To define nodes with a **nodedef** file, issue the **definenode** command, on the management server, and specify the full path name of the **nodedef** file. For example:

```
definenode -f /tmp/nodedef
```

Using the command line to specify nodes

To define a node's hardware attributes from the command line, issue the **definenode** command, followed by the attributes you want to assign. We used the sample of a completed Hardware Node Attributes Template, from the *CSM for Linux: Hardware Planning and Control Guide* to create the example below.

Setting a node's *HWControlNodeID* attribute to the short host name of the node can simplify the node definition process. The following **definenode** command example defines the node short host names as the *HWControlNodeID* attribute values. If the *HWControlNodeID* attribute values were not set to the node short host names, then the **nodedef** file would be used to specify each attribute value. If a node's *HWControlNodeID* is the short host name, then the following command can be run once to define all nodes attached to the hardware control point:

```
definenode -n clsn01.pok.ibm.com -c 20 -H mgtn03.pok.ibm.com:10,- mgtn04.pok.ibm.com:10 \
-C mgtn02.pok.ibm.com:1:0:16,mgtn05.pok.ibm.com:2:0:16 PowerMethod=netfinity ConsoleMethod=esp
```

All *HWControlNodeID* node attribute values attached to a hardware control point must be unique. For Netfinity xSeries 330 and 342 nodes, the *HWControlNodeID* value must match the Netfinity text ID set in the hardware. If the *HWControlNodeID* values are changed to the short host names of the nodes, then the **systemid** command must be subsequently run to correctly set the new user ID and password information in the CSM database.

See the man page or *CSM for Linux: Administration Guide* for details on **definenode** command line syntax and more examples of using the command.

Step 6. Verify the node definitions

After **definenode** has run, the management server has been set up with all the node information for CSM, and you are now ready to verify the node definitions. This section tells you how to verify and customize the cluster node definitions before the nodes are installed. Since the actual node installation has not happened yet, you can make changes to any of the node definitions at this time.

To determine whether the node has been defined, issue the **lsnode** command:

```
lsnode
```

The system responds with a list (output) for each node that was successfully defined. If a node has not been defined, it will not appear in the output for **lsnode**.

To display all the information about each node, use the **lsnode** command with the –l (lowercase l, not uppercase i) option:

```
lsnode -l
```

The system responds with a line, containing extended information, for each node that was successfully defined. If a node has not been defined, it will not appear in the output for **lsnode**.

Note: If a node has not been defined, it will not be installed.

If something needs to be corrected, you can either remove the node that was not successfully defined and then rerun the **definenode** command with the correct arguments, or you can make the necessary changes to the attributes of the node. Note that all of the attributes for a node might have null values at this point.

To remove a node before redefining it, use the **rmnode** command:

```
rmnode hostname
```

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

```
chnode hostname attr=value
```

See the man pages or the *CSM for Linux: Administration Guide* for more information about the **lsnode**, **chnode**, and **rmnode** commands.

Step 7. Modify the Kickstart node configuration template (optional)

In order to proceed with the installation, you will need to provide configuration data for the nodes. Kickstart provides a basic node configuration template that you can either use as it is, or modify. The template file is located on your system at **/opt/csm/install/kscfg.tmpl.RedHat.***Version*. For example: **/opt/csm/install/kscfg.tmpl.RedHat.7.2**. For your convenience, a sample Kickstart node configuration template is also contained in this book, in Appendix D, “Sample Kickstart configuration template files” on page 39.

The Kickstart configuration template file contains the following information:

- Initial root password (defaults to *cluster*), plus an additional user ID called *admin* with a password of *cluster*.
- Number of disks installed
  - Type of each disk
  - Size of each disk
  - Whether to clear partitions
- Disk partition table

  A sample disk partition table is provided, which can be modified.

- RPMs to install

  A standard list is provided; you can modify the list, or you can use your own list.
- Language and time zone
- Post-install script

  This script does standard set up and CSM-specific set up. The script may be modified to suit your installation, provided the marked sections are not altered.

Before updating the Kickstart configuration template, it is recommended that you make a backup copy of the original file, and give it a different name. The template file you use must retain the original name, and be stored in its original directory, so you will need to update the original Kickstart configuration template file.

As stated earlier, you can use the Kickstart node configuration template as it is or modify it. See the annotations in the **kscfg.tmpl** itself for instructions on how to modify the template. Changes made to the template file affect the entire cluster and should be made before going to the next step.

Step 8. Validate rpower

Before installing CSM on the nodes, test **rpower** to make sure it is configured. To test **rpower**, issue the following:

```
rpower -a query
```

The output from the **rpower** command lists of all the nodes and their status. For example:

```
root@c5n71:/u/build # rpower -a query
c5n72.ppd.pok.ibm.com on
c5n73.ppd.pok.ibm.com on
c5n74.ppd.pok.ibm.com off
c5n75.ppd.pok.ibm.com off
```

Step 9. Create the Kickstart node configuration files

After you have prepared your Kickstart configuration template, you use the **csmsetupks** command to create a Kickstart configuration file for each of the nodes in your cluster. **csmsetupks** uses the information from your Kickstart node configuration template to create the Kickstart configuration files. **csmsetupks** does the following:

- Copies Red Hat CD-ROMs to the **/csminstall** directory on the management server
- Creates an **/etc/dhcpd.conf** file
- Collects MAC addresses for all nodes
- Starts the DHCP, NFS, and ATFTP daemons
- Creates the files necessary for the network boot into **/csminstall**
- Generates a KickStart configuration file for each node.

To run the **csmsetupks** command for all the premanaged nodes of your cluster, type the following:

```
csmsetupks -P
```

csmsetupks saves the output to a log file called **/var/log/csm/csmsetupks.log**.

Step 10. Verify the Kickstart node configuration

To verify the Kickstart configuration, look at the **/etc/dhcp.conf** file to make sure there is a static entry for each node in the cluster.

Note that you can modify the Kickstart configuration file that was generated for each node by **csmsetupks**. The Kickstart configuration file that is generated for each node is called **/csminstall/csm/***CSMVersion***/kickstart.Linux.***Version/node-ip-address***-kickstart**. For example: **/csminstall/csm/1.2.0/kickstart.Linux7.1/9.117.241.32–kickstart**. Modifying this file affects only the settings on this node and should be done after running **csmsetupks**.

Step 11. Install CSM and the operating system on the nodes

Nodes can be in one of three *Modes* at any given time during installation. When a node has only been defined, but not actually added to the cluster, it is in **PreManaged** mode. While the node is being installed (while the **installnode** command is running), its Mode is **Installing**. And finally, after a node has been defined and added to the cluster with the **installnode** command, its Mode is **Managed**.

The **installnode** command is run on nodes that have already been defined to the system. When **installnode** is run, each node that was just defined (whose *Mode* attribute is **PreManaged**) becomes a managed node of the cluster. Note that if the **installnode** command fails, the node's Mode attribute remains **PreManaged**.

To install the premanaged nodes of the cluster, using the **installnode** command, type the following:

```
installnode -P
```

installnode runs asynchronously. In other words, immediately after the installation process is initiated (the node is rebooted), **installnode** exits, even though the installation may not be complete.

In addition to installing CSM on the nodes, **installnode** also installs the operating system, using Linux Kickstart. The appropriate software, listed in "Understanding CSM software requirements" on page 1 is installed automatically by the **installnode** command on the nodes, if it is not already installed, or if it is installed at a previous level.

After a full installation, the boot order in the BIOS of the node can be left as it is: diskette, CD-ROM, network, hard disk. In this case, every time the node boots, it uses **dhcp** (Dynamic Host Configuration Protocol) to contact the management server, which uses **pxelinux** to boot the node from its hard drive. Or, after the full installation is complete, you can change the boot order in the BIOS back to: diskette, CD-ROM, hard disk, network.

Step 12. Verify the installation

For a full installation, you can use the **rconsole** command to view the progress of the installation on each node. To run the **rconsole** command on all the nodes, type the following:

```
rconsole -a
```

The **rconsole** command displays a console for each node in the cluster.

To check the results of the installation, you can use the **monitorinstall** command. The monitorinstall command starts the installation monitor tool, which displays the status of the installation on each of the nodes. It returns the following types of status:

- installed
- installing
- not installed
- failed install

To run the **monitorinstall** command, type the following:

```
monitorinstall
```

The monitorinstall command writes output similar to this:

```
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      Installed
clsn07.ppd.pok.ibm.com      Installed
clsn08.ppd.pok.ibm.com      Not Installed
```

All nodes should be listed as *Installed*.

If there is a problem, see **/var/log/csm/installnode.log** on the management server or **/var/log/csm/install.log** on the managed node for information on what happened on each node during installation.

To check the installation of the nodes, you can also use the **lsnode** command, as follows:

```
lsnode -a Mode
```

lsnode returns a list of the active nodes in the cluster, along with their *Mode* attributes. Nodes that were successfully installed are set to **Managed**.

To check that dsh is working, issue a simple **dsh** command. For example:

```
dsh -a date
```

A list of the nodes in the cluster is returned, with a date for each node.

Chapter 5. CSM post-installation tasks

Getting started with the newly installed cluster

This section tells you how you can determine whether the installation was successful. It also gives you some suggestions on how to get started using Cluster Systems Management. After installation is successfully completed, remote RMC and CSM commands are enabled. To verify that the installation was successful, follow the directions in the sections below.

Verifying that dsh is working on the nodes

To verify that **dsh** is working on all of the nodes, run the **dsh** command with the **date** option, as follows:

```
dsh -a date
```

A list of nodes with the date on each node is returned.

Checking the status of the nodes

To see the status of all the nodes, you can use the **monitorinstall** , **lsnode**, or **nodegrp** command. To use the **monitorinstall** command, type:

```
monitorinstall
```

The **monitorinstall** command writes output similar to this:

```
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      Installed
clsn07.ppd.pok.ibm.com      Installed
clsn08.ppd.pok.ibm.com      Not Installed
```

All nodes should be listed as *Installed*.

To see the installation status of all the nodes in your cluster, using the **lsnode** command, type:

```
lsnode -a Mode
```

The result shows you the *Mode* of each node. The mode for all the nodes should be **Managed**.

To see the installation status of a group of nodes, using the **nodegrp** command, type:

```
nodegrp ManagedNodes
```

The command displays a list of all the nodes in the cluster that are considered managed (defined and installed).

Verifying that RMC is working

To verify that RMC is working, use the **lsnode** command, as follows:

```
lsnode -H
```

This command retrieves information about the attributes from each node.

Verifying the power status of the nodes

To verify the power status of the nodes (whether they are on or off), type:

```
rpower -a query
```

A list of nodes with their associated power state is returned.

Verifying reachability of the nodes

To verify whether the nodes are reachable, type:

```
lsnode -p
```

The ping status of the nodes is returned.

Getting list of predefined conditions, responses, and dynamic node groups

CSM provides a set of predefined conditions, responses, and dynamic node groups. To see a list of the predefined conditions, use the RSCT **lscondition** command, as follows:

```
export CT_MANAGEMENT_SCOPE=1
lscondition
```

To see a list of the predefined responses, use the RSCT **lscondresp** command, as follows:

```
export CT_MANAGEMENT_SCOPE=1
lscondresp
```

To see a list of the predefined dynamic node groups use the **nodegrp** command, as follows:

```
nodegrp
```

Getting started with the Configuration File Manager

To begin working with the Configuration File Manager, use the following example. The example sets up the **cfmupdatenode** command to run whenever the **/cfmroot/tmp/myfile** file has changed, distributing the changed file across all nodes in the cluster.

```
export CT_MANAGEMENT_SCOPE=1
startcondresp "CFMRootModTimeChanged" "CFMModResp"
mkdir /cfmroot/tmp
touch/tmp/myfile
cp /tmp/myfile /cfmroot/tmp/myfile
```

For more information on the Configuration File Manager, see *CSM for Linux: Administration Guide*.

Getting started with monitoring

To try out monitoring, use the following example.

- To set up a condition that causes a wall message to be broadcast on the management server when a node is shut down, type:

  ```
  export CT_MANAGEMENT_SCOPE=1
  startcondresp NodeReachability BroadcastEventsAnyTime
  ```

- To set up node groups called, for example, **servers** and **admin**, and then run the **vmstat** command on them, do the following:
  1. To form a node group called **servers** that consists of nodes c5bn07, c5bn08, c5bn09, c5bn10, and c5bn11, type:

     ```
     nodegrp -a c5bn07,c5bn08,c5bn09,c5bn10,c5bn11 servers
     ```

  2. To form a node group called **admin** that consists of nodes c5bn12 and, c5bn13, type:

     ```
     nodegrp -a c5bn12,c5bn13 admin
     ```

  3. To display the the node groups that have been defined, type:

     ```
     nodegrp
     ```

     The output is similar to:

```
admin
servers
```

4. To display the nodes in the node group **servers**, type:

   ```
   nodegrp servers
   ```

   The output is:

   ```
   c5bn07.ppd.pok.ibm.com
   c5bn08.ppd.pok.ibm.com
   c5bn09.ppd.pok.ibm.com
   c5bn10.ppd.pok.ibm.com
   c5bn11.ppd.pok.ibm.com
   ```

5. To run the **vmstat** command on the node group **servers** and have formatted output, type:

   ```
   dsh -N servers vmstat | dshbak
   ```

   The output is similar to:

   ```
   HOST: c5bn08.ppd.pok.ibm.com
   ----------------------------
      procs                      memory    swap          io     system         cpu
    r  b  w   swpd   free   buff  cache  si  so    bi    bo   in    cs  us  sy  id
    0  4  1 442440 192576  56292 635808   0   0     0     0    1     1   0   0   0

   HOST: c5bn09.ppd.pok.ibm.com
   ----------------------------
      procs                      memory    swap          io     system         cpu
    r  b  w   swpd   free   buff  cache  si  so    bi    bo   in    cs  us  sy  id
    0  4  1 423692 214232  56240 615396   0   0     0     0    1     1   0   0   0

   HOST: c5bn10.ppd.pok.ibm.com
   ----------------------------
      procs                      memory    swap          io     system         cpu
    r  b  w   swpd   free   buff  cache  si  so    bi    bo   in    cs  us  sy  id
    0  4  1 405904 162404  56248 604424   0   0     0     0    4     1   0   0   1

   HOST: c5bn11.ppd.pok.ibm.com
   ----------------------------
      procs                      memory    swap          io     system         cpu
    r  b  w   swpd   free   buff  cache  si  so    bi    bo   in    cs  us  sy  id
    0  4  1 443564 135240  56212 636256   0   0     0     0    4     1   0   0   1
   ```

Understanding the installation and configuration log files

Several of the commands you run during the installation process create log files, which are placed in the **/var/log/csm** directory. These log files contain extended error and informational messages from the command.

The following log files are created:

Table 1. Log files that are created during installation

Source	Log file	Location
installms command	**/var/log/csm/installms.log**	Management server
installnode command	**/var/log/csm/installnode.log**	Management server
updatenode command	**/var/log/csm/updatenode.log**	Management server
installnode and **updatenode** commands result in data being written to this log	**/var/log/csm/install.log**	Cluster node
csmsetupks	**/var/log/csm/csmsetupks.log**	Management Server

Note that each log directory can contain up to five versions of the log, each named with a suffix that indicates its relative age. For example, the newest **installms** log file is called *installms.log*, while the oldest is called *installms.log.4*. When the next log file is created, the file that was called *installms.log* becomes *installms.log.1* to make way for the newly created *installms.log*, and the file that was *installms.log.4* is deleted.

Adding a node to an existing cluster

You can add nodes to the cluster by running **definenode**, then **csmsetupks** and **installnode** (for a full install) or **updatenode** (for a CSM-only install). For more information about performing a full installation, see Chapter 4, "Installing CSM and the operating system on the nodes" on page 15. For more information about performing a CSM-only installation, see Chapter 3, "Installing only CSM on the cluster nodes" on page 9.

Removing a node from an existing cluster

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. To remove a node from the cluster, type:

```
rmnode hostname
```

A removed node can be added back into the cluster by running **definenode**, then **updatenode**.

Reinstalling CSM on a node

The procedure for reinstalling a CSM node is different, depending on whether you want to reinstall CSM only, or if you want to reinstall CSM and the operating system.

Reinstalling CSM only

To reinstall CSM, do the following:

- Issue the **updatenode** command to reinstall and configure CSM on all the defined nodes:

  ```
  updatenode -n nodename
  ```

See the man pages or the commands chapter of the *CSM for Linux: Administration Guide* for more information about the **updatenode** and **definenode** commands.

Reinstalling CSM and Linux

To reinstall both CSM and the operating system, do the following:

1. If you need to change one or more of the node attributes, such as InstallDistributionVersion (operating system version), use the **chnode** command. For example, to change the operating system version from Red Hat 7.1 to 7.2, you would use the command like this:

   ```
   chnode -N MangagedNodes InstallDistributionVersion='7.2'
   ```

   For more information about the **chnode** command, see the man page or the commands chapter of *CSM for Linux: Administration Guide*
2. Run the **csmsetupks** command to reconfigure Red Hat Kickstart and create the Kickstart configuration files.
3. Run the **installnode** command to reinstall the nodes with the new operating system level.

Uninstalling CSM

At the time of this writing, the **rpm** command does not reorder packages based on dependencies when doing an uninstall. Instead, the packages are uninstalled in the order in which they are specified on the command line. To properly uninstall the CSM and RSCT packages and the packages on which they depend, the packages should be specified as follows.

Uninstalling the management server

To uninstall the management server on the **Red Hat Linux 7.1** operating system:

```
rpm -e csm.server csm.gui.dcem csm.dsh csm.core rsct.core rsct.core.utils src
```

To uninstall the management server on the **Red Hat Linux 7.2** operating system:

```
rpm -e csm.server csm.gui.dcem csm.dsh csm.core src rsct.core.utils rsct.core
```

Uninstalling a node

To uninstall a node on the **Red Hat Linux 7.1** operating system:

```
 rpm -e csm.client csm.core rsct.core rsct.core.utils src
```

To uninstall a node on the **Red Hat Linux 7.2** operating system:

```
 rpm -e csm.client csm.core src rsct.core.utils rsct.core
```

If packages that depend on the CSM and RSCT packages are uninstalled with the same invocation of the **rpm** command, those packages should be specified before the CSM and RSCT packages.

Chapter 6. Migrating to the latest level of CSM

This chapter provides migration instructions for CSM 1.2. Migrating is the task of upgrading a previous level of CSM to the most current.

Note: CSM does not support migrating from one level of Red Hat Linux to another. This chapter only describes migrating CSM at the Red Hat Linux 7.1 level.

Migrating from CSM 1.1 to CSM 1.2

If you are currently using CSM 1.1, you will need to follow the procedure in this section to move to the CSM 1.2 release.

Step 1

Mount the CSM 1.2 CD-ROM and run the **savenodedata1.1** command, as follows:

```
mount /dev/cdrom /mnt/cdrom
/mnt/cdrom/savenodedata1.1 > /tmp/nodedef.migrate
```

The **savenodedata1.1** command generates the definitions for the nodes that are currently in your cluster and redirects them to a file called **nodedef.migrate**. Later in this procedure, you will use the **nodedef.migrate** file to provide your current node definitions to the new level of CSM.

Step 2

Remove all CSM 1.1 RPMs from the nodes as follows:

```
rpm -e csm.client csm.core rsct.core rsct.core.utils src
```

Step 3

Uninstall all of the CSM 1.1 and RSCT code as follows:

```
rpm -e csm.gui.dcem csm.dsh csm.server csm.core rsct.core rsct.core.utils src
```

Note: cfengine was used by CSM 1.1, but its function is replaced in CSM 1.2 with rdist. Note that because you might be using cfengine for other things, cfengine was not removed when you uninstalled CSM. However, you can manually remove cfengine if you don’t plan on using it.

Step 4

Create a partition called **/csminstall**. See “Step 2. Create the /csminstall partition” on page 5 for information on how to do this.

Step 5

Install the new version of the **csm.core** package from the CSM 1.2 CD-ROM. See “Step 4. Install the csm.core file set” on page 5 for information on how to do this.

Step 6

Install the remainder of CSM 1.2 and RSCT by running the **installms** command. See “Step 5. Install the CSM software” on page 5 for information on how to do this.

Step 7

Redefine the nodes of your cluster for CSM 1.2, using the **definenode** command and the **nodedef.migrate** file, as follows:

```
definenode -f /tmp/nodedef.migrate
```

Note that some of the node attributes that are used in CSM 1.2 were not present in CSM 1.1. When you run the **definenode** command, above, the missing attributes are filled in with defaults from the management server or left blank, as needed.

Step 8

For CSM-only installations, upgrade to the latest level of CSM and RSCT, using the **updatenode** command, as follows:

```
updatenode -P
```

Or, for a full installation, if you wish to copy the existing data in **/tftpboot** to the new **/csminstall** directory, perform the following steps:

1. Run the **csmsetupks** command, using the –p flag to point to the **/tftpboot** directory, as follows:

   ```
   csmsetupks -p /tftpboot/rh71
   ```

   Note also that if you customized the Kickstart configuration files when you installed CSM 1.1, those configurations will be lost when you install CSM 1.2. As a result, you will need to customize the new configuration files. The CSM 1.1 Kickstart configuration files were located in **/tftpboot/ks71**.
2. If you wish to recover file system space, do the following:

   ```
   rm -rf /tftpboot/rh71
   ```
3. Run the **installnode** command:

   ```
   installnode -P
   ```

Changes to CSM commands

The following commands have been removed from CSM since the last release:

- **addnode** – now use **definenode –U**
- **cquery**
- **createnode** – now use **definenode**

The following commands have been renamed since the last release:

- **cforce** — new name is **cfmupdatenode**
- **setupks** — new name is **csmsetupks**

The following commands have been added to CSM since the last release. See the Commands chapter of *CSM for Linux: Administration Guide* for information about each of these commands.

- **updatenode**
- **savenodedata1.1**

Note that there is a new man page called **nodeattributes**, which lists the node definition attributes. This is not a command but rather an informational man page.

Changes to CSM command options

The following attributes have been removed:

- The **–P** option has been removed from the **rmnode**, **chnode**, **lsnode**, **rpower**, **rconsole**, **nodegrp**, and **dsh** commands. Note that the –P option can only be used with the **updatenode**, **installnode**, and **csmsetupks** commands.
- The **–B** option has been removed from all CSM commands.
- The **–i**, **–p**, and **–t** options have been removed from the **dmsctrl** command.

Coexistence

Within a cluster, the version of CSM that is installed on the management server must be at the same level as the version of CSM that is installed on the nodes.

The version of the Red Hat Linux operating system that is installed on the management server and nodes of your cluster may be at either 7.1 or 7.2. There is no requirement that the management server be at the same level, or a later level, than the version of Red Hat that is installed on the nodes.

Chapter 7. Backing up and restoring CSM data

The chapter discusses how to back up CSM directories and how to restore CSM data. It also tells you how to backup and restore **/var/ct**.

Backing up CSM data

The following directories and all their subdirectories should be backed up periodically. These directories are used by Cluster Systems Management to store persistent data and other configuration information. In addition, if major configuration changes are made by means of CSM commands, a back up should be performed.

These directories only need to be backed up on the management server.

- CSM Configuration Data: **/etc/opt/csm**
- CSM Runtime Data: **/var/opt/csm**
- RMC Persistent Data: **/var/ct.**. To backup **/var/ct**, do the following:
  1. Run the following command to stop the RMC subsystems.

     ```
     /usr/sbin/rsct/bin/rmcctrl -z
     ```

     The command does not complete until all RMC subsystems are stopped.
  2. Perform a backup of the **/var/ct** directory
  3. Run the following command to restart the RMC subsystems.

     ```
     /usr/sbin/rsct/bin/rmcctrl -s
     ```

You can also backup CSM data by using the **lsnode** command as follows:

```
lsnode -s > nodedef
```

Restoring CSM data

Do the following procedure to restore CSM data:

1. Reinstall CSM if necessary.
2. Copy **/etc/opt/csm** and **/var/opt/csm** from the backup media.
3. Refresh RMC, as follows:
   a. Stop the RMC subsystems by running the following command:

      ```
      /usr/sbin/rsct/bin/rmcctrl -z
      ```

      The command does not complete until all RMC subsystems are stopped.

   b. If in HA cluster mode, run the following command to stop the cluster subsystems:

      ```
      /usr/sbin/rsct/bin/cthactrl -k
      ```

   c. Restore the **/var/ct** directory.

   d. If in HA cluster mode, run the following command to start the cluster subsystems:

      ```
      /usr/sbin/rsct/bin/cthactrl -s
      ```

   e. To restart the RMC subsystems, run the following command:

      ```
      /usr/sbin/rsct/bin/rmcctrl -s
      ```

Appendix A. Node attributes templates

Hardware Node Attributes Template

This section contains a blank Hardware Node Attributes Template for your use. Duplicate the template, as needed, for your cluster. Next, refer to *CSM for Linux: Hardware Planning and Control Guide* for information about the required hardware node attributes. The *CSM for Linux: Hardware Planning and Control Guide* also includes an example of a template that has been completed.

Table 2. Hardware Node Attributes Template

Hostname	*HWControlPoint*	*Power Method*	*HWControl NodeId*	*ConsoleServerName*	*Console Server Number*	*Console Method*	*Console PortNum*

Software Node Attributes Template

Note that for the typical system, the software attributes will be the same for each node. If this is the case with your system, the attributes will be simple to understand, and filling out this template may not be necessary. However, if the nodes of your system have different software attributes, it may be difficult to keep track of them so, in this case, you should complete this template. It will be useful for future reference, especially during installation.

Completing the Software Node Attributes Template

Before starting, note that the Software Node Attributes Template accommodates 25 nodes. If you have more than 25 nodes in your cluster, make copies of the template as needed.

For each node (*Hostname*) in the cluster, enter the following information on the template:

Hostname
: Specifies the host name of the node. For example, **node1**.

InstallCSMVersion
: Specifies the version, release, and modification of CSM that will be installed on the node. Specify as *version.release.modification*. For example, **1.2.0**.

InstallOSName
: Specifies the operating system type. Currently, the only valid value is for this attribute is **Linux**.

InstallDistributionName
: Specifies the operating system distribution. Currently, the only valid value for this attribute is **RedHat**.

InstallDistributionVersion
: Specifies the operating system version. Specify as *version.release*. The valid values for this attribute are **7.1** or **7.2**.

InstallPkgArchitecture
: Specifies the machine architecture of the node. Currently, the only valid value for this attribute is **i386**, but the i386 value encompasses i486, i586, and i686 architectures.

Table 3. Software Node Attributes Template

Hostname	*InstallCSMVersion*	*InstallOSname*	*InstallDistributionName*	*InstallDistributionVersion*	*InstallPkgArchitecture*

Appendix B. Understanding the installation directories and files

The **installms** command establishes the CSM directories and the required packages when it is run from the management server. The figure below illustrates the overall directory structure of CSM after it has been installed on your system.

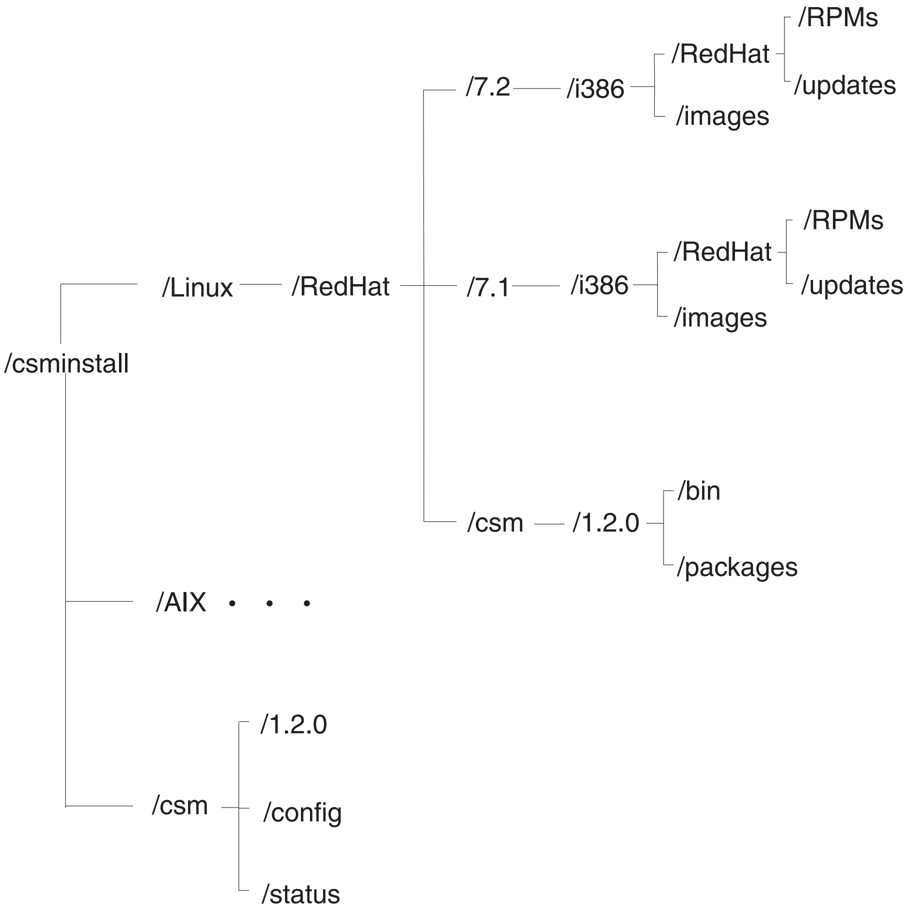

Figure 1. CSM installation directory structure

The table below describes some of the directories that may be of interest to you. Note that the directory names may contain the following variables, which correspond to ManagedNode attributes:

- *InstallOSName* — Represents the operating system type. For example, /csminstall/**Linux**.
- *InstallDistributionName* — Represents the operating system distribution (*RedHat* is the only valid value). For example, /csminstall/Linux/**RedHat**.
- *InstallCSMVersion* — Represents the version, release, and modification of CSM (specify as *version.release.modification*). For example, /csminstall/Linux/RedHat/csm/**1.2.0**.
- *InstallDistributionVersion* — Represents the operating system version (specify as *version.release*). For example, /csminstall/Linux/RedHat/**7.2**.
- InstallPkgArchitecture — Represents the machine architecture. For example, /csminstall/Linux/RedHat/7.2/**i386**. Note that while the variable and directory name is **i386**, your machine architecture may actually be i486, i586, or i686. The default *InstallPkgArchitecture* attribute for any of these architectures is **i386**.

Table 4. CSM installation directories

Directories/Files	Description
/csminstall	Main (parent) CSM install directory. **/csminstall** is mounted on the cluster nodes during both CSM-only and full installations.
/csminstall/csm	Contains tools and libraries required for CSM (for files that are not specific to a particular version of CSM).
/csminstall/csm/config	Contains temporary node configuration files. Do not modify the contents of this directory.
/csminstall/csm/status	Status files of node installation.
/csminstall/csm/_InstallCSMVersion_	Contains tools and libraries required for CSM (for files that are specific to a particular version of CSM).
/csminstall/csm/_InstallCSMVersion_**/kickstart.**_InstallDistributionNameInstallDistributionVersion/node-ipaddr_-kickstart	The Kickstart configuration file for each node.
/csminstall/_InstallOSName_	Contains files needed to install and configure nodes based on the operating system type.
/csminstall/_InstallOSName/InstallDistributionName_**/csm/**_InstallCSMVersion_**/packages/*.rpm**	The directory in which the CSM RPM packages that will be installed on the management server and the nodes are stored. **installms** copies the CSM RPMs from the CSM CD-ROM to this directory. RPMs of open source and IBM Java2 programs also get stored here.
/csminstall/_InstallOSName/InstallDistributionName/InstallPkgArchitecture_**/RedHat/RPMS/*.rpm**	The directory in which the RPM packages, that will be installed on the management server and the nodes, are stored. **installms** copies the RPMs that are needed to install CSM on the management and the nodes to this directory. **csmsetupks** copies the RPMs needed to install the operating system to this directory.
/etc/dhcpd.conf	The system configuration file for the dhcp server.
/opt/csm/install/kscfg.tmpl.RedHat._InstallDistributionVersion_	The Kickstart configuration template file.

Appendix C. Installing program updates

Program updates come to you as RPM packages. You can update the version of CSM on either the management server or a node. Note, however, that the version of CSM on the management server must always be at the same, or more recent level than any other version of CSM installed on a cluster node.

You can install any package that has a version level of 1.2.0.* (where * can be any value). To determine the version of a particular package that is installed on your system, use the **rpm** command **(rpm –q** *packagename*). For example:

```
rpm -q csm.core
```

Updating the management server

To apply program updates to the management server, issue the **installms** command with the **–p** (path) flag. The **–p** flag is used to indicate the location of the updated package. For example,

```
installms -p /csminstall/Linux/RedHat/csm/1.2.0/packages
```

Updating a node

To apply program updates to a node or group of nodes, issue the **updatenode** command. The **updatenode** command always installs the latest level of CSM (from the management server). For more information on using the **updatenode** command, see *CSM for Linux: Administration Guide*

```
updatenode -n nodename
```

To install program updates on remote nodes, use the **dsh** command.

Appendix D. Sample Kickstart configuration template files

This appendix contains sample Kickstart configuration template files for both RedHat 7.1 and RedHat 7.2. If you are using RedHat 7.1, see “RedHat 7.1 Kickstart configuration template file”. If you are using RedHat 7.2, see “RedHat 7.2 Kickstart configuration template file” on page 48.

Note that there are particular variables in the Kickstart configuration template and the generated file in the format #VARIABLE# that must not be deleted. Many of these are automatically customized with the appropriate values during the process of generating the Kickstart configuration file. For example, the following are some of the variables that are automatically customized:

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

#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 (**/csminstall/Linux/RedHat/***Version/MachineArchitecture*). For example, **/csminstall/Linux/RedHat/7.1/i386**.

RedHat 7.1 Kickstart configuration template file

The following is a sample Kickstart configuration template for RedHat 7.1. The template file is located in **/opt/csm/install/kscfg.tmpl.RedHat.7.1**. The most commonly updated sections of this template are shown in boldface type.

```
#
#Kickstart Configuration File for RedHat 7.1 Only
#

lang en_US
keyboard "us"

network --bootproto dhcp

#
# Where are the RedHat RPMs?
# nfs --server hostname.of.server --dir /path/to/RH/CD/image
#
# If your RedHat RPMs are not on the management server in the normal location,
# you can replace the NFS_HOSTNAME and NFS_DIR variables with your own
# values.
#
nfs --server #NFS_HOSTNAME# --dir #NFS_DIR#

#
# Clear the MBR
#
zerombr yes

#
# Wipe out the disk
#
clearpart --all

#
# Disk partition table. Customize it to fit your needs.
# /boot is strongly recommended
#
part /boot --size 50   --fstype ext3
part /     --size 1024 --fstype ext3
part swap  --size 1024 --ondisk sda
part /var  --size 1024 --fstype ext3
part /opt  --size 512  --fstype ext3
part /usr  --size 2048 --fstype ext3
#
# install or upgrade
#
install

#
# mouse selection
#
#mouse genericps/2 --emulthree
mouse none

#
# Select a time zone
# Add the --utc switch if your hardware clock is set to GMT
# Use the /usr/sbin/timeconfig tool to determine valid values.
#
# The timezone is set by default to the same timezone that the management
# server has.  This is found in /etc/sysconfig/clock on the management server.
#
timezone #TIMEZONE_UTC# #TIMEZONE#

#
# Don't set up X Windows.
#
skipx

#
# To generate an encrypted root password use:
#
# perl -e 'print crypt("blah","Xa") . "\n";'p
# openssl passwd -apr1 -salt xxxxxxxx password
#
# where "blah" is your root password.
#
# --iscrypted is broken
#
#rootpw --iscrypted XaLGAVe1C41x2
#rootpw XaLGAVe1C41x2 --iscrypted
rootpw cluster
auth --useshadow --enablemd5

#
# Lilo
#
lilo --location mbr

#
# Reboot after installation
#
reboot

#
#end of section
#

#
# Choose the packages and package groups to install
# The following files must not be removed, or else csm will not work:
#   @ Networked Workstation
#   pdksh
#   kernel-headers
#   rsh-server
#   nfs-utils
#   xinetd
#   tftp
#   binutils
#   others ????
#
#
%packages
@ Networked Workstation
compat-libstdc++
kernel-smp
kernel-headers
glibc-devel
binutils
cpp
gcc
flex
autofs
pdksh
tcsh
ntp
xosview
XFree86
tftp
xinetd
rsh
rsh-server
telnet-server
pvm
pvm-gui
psacct
nfs-utils
raidtools
ucd-snmp
rsync
yp-tools
ypserv
ypbind
m4
sendmail-cf
%pre#
# Pre Install script
#
# This script may be modified as needed.  However, certain sections should
# not be altered or else the CSM installation will not work correctly. These
# sections are marked with the text "DO NOT ERASE THIS SECTION".
#

###################### DO NOT ERASE THIS SECTION (begin) ####################
#
# Set some variables first.  These get filled in by the management server.
export NODE_HOSTNAME=#NODE_HOSTNAME#
export NFS_IP=#NFS_IP#
export TFTP_IP=#TFTP_IP#
export CSM_VERSION=#CSM_VERSION#
export CSMINSTALL=/csminstall
export ATFTP=$CSMINSTALL/csm/$CSM_VERSION/atftp
export LOGFILE=/var/log/csm/install.log

write_log()
{
    mkdir -p /var/log/csm
    echo "`date`: $*" >> $LOGFILE
} # END write_log

LOGGER='write_log'

$LOGGER "`mkdir $CSMINSTALL 2>&1`"
$LOGGER "`mount -o nolock $NFS_IP:$CSMINSTALL $CSMINSTALL 2>&1`"
```

```
# Update the status file on the management server to say that the install is
# starting.

msg="Reboot complete"
/csminstall/csm/write_status -v -p $ATFTP -S $TFTP_IP -n $NODE_HOSTNAME $msg

msg="Starting Kickstart Install"
/csminstall/csm/write_status -v -p $ATFTP -S $TFTP_IP -n $NODE_HOSTNAME $msg

umount $CSMINSTALL

###################### DO NOT ERASE THIS SECTION (end) ######################
exit 0

%post
#
# Post Install script
#
# This script may be modified as needed.  However, certain sections should
# not be altered or else the CSM installation will not work correctly. These
# sections are marked with the text "DO NOT ERASE THIS SECTION".
#

###################### DO NOT ERASE THIS SECTION (begin) ####################
#
# The LOGGER variable is used to send items to some kind of log.  It can be
# set to 'write_log' in order to call the 'write_log' function (defined below)
# which sends the log output to /var/log/csm/install.log on the node.
# If LOGGER is set to /usr/bin/logger, the Linux syslog facility is used
# instead.
#
#export LOGGER=/usr/bin/logger
export LOGGER=write_log

#
# Set variables
#
# The following variables are substituted into the kickstart template file
# from the setupks tool
#
# MGMTSVR_HOSTNAME   = management server (Long Hostname)
# MGMTSVR_SHORTHOST  = management server (Short Hostname)
# MGMTSVR_IP         = management server (IP)
# NODE_HOSTNAME      = node hostname (Long Hostname)
# NODE_HEX           = node IP address in hexidecimal format
# NFS_HOSTNAME       = NFS Server (Long Hostname) - same as MGMTSVR_HOSTNAME
# NFS_IP             = NFS Server (IP) - same as MGMTSVR_IP
# NFS_DIR            = Path to RedHat CD image on NFS Server
# TFTP_IP            = TFTP Server (IP) - same as MGMTSVR_IP
# NAMESERVERS        = list of comma delimited name servers(IPs or Hostnames)
# DNSDOMAIN          = DNS domain name of the node(NAME)
# SETUP_RSH          = Set to "Y" if $DSH_REMOTE_CMD variable on the
#                      management server is set to "rsh" or is blank.
# CSM_VERSION        = The version of CSM that is being installed
# TIMEZONE           = Timezone string of the Management Server
# TIMEZONE_UTC       = Timezone Universal System Time Setting
#                      on the Management Server (either --utc or blank)
#
export KICKSTART_LOGFILE=/var/log/csm/install.log
export MGMTSVR_HOSTNAME=#MGMTSVR_HOSTNAME#
export MGMTSVR_SHORTHOST=#MGMTSVR_SHORTHOST#
export MGMTSVR_IP=#MGMTSVR_IP#
export NODE_HOSTNAME=#NODE_HOSTNAME#
export NODE_HEX=#NODE_HEX#
export NFS_HOSTNAME=#NFS_HOSTNAME#
export NFS_IP=#NFS_IP#
export TFTP_IP=#TFTP_IP#
export NAMESERVERS=#NAMESERVERS#
export DNSDOMAIN=#DNSDOMAIN#
export SETUP_RSH=#SETUP_RSH#
export CSM_VERSION=#CSM_VERSION#

#
# Hardcode some other variables
#
export GATEWAY=$(netstat -rn | grep 0.0.0.0 | tail -1 | awk '{print $2}')
export CSM_MANPATH=/opt/man
export CSM_PATH=/opt/bin
export TFTPBOOT=/tftpboot
export CSMINSTALL=/csminstall
export ATFTP=$CSMINSTALL/csm/$CSM_VERSION/atftp

#
# Function to add some text to the install.log file on the node.
#
write_log()
{
    mkdir -p /var/log/csm
    echo "`date`: $*" >> $KICKSTART_LOGFILE
} # END write_log

#
# Use atftp to get a file from the management server.  The remote file is
# relative to /tftpboot
#
atftp_get()
{
    _remote_file=$1
    _local_file=$2

    # Use this format if LOGGER=/usr/bin/logger
    #echo "get $_local_file $remote_file" | $ATFTP $TFTP_IP 2>&1 | $LOGGER

    # Use this format if LOGGER=write_log
    $LOGGER "`echo \"get $_remote_file $_local_file\" | $ATFTP $TFTP_IP 2>&1`"
} # END atftp_get

#
# Use atftp to put a file to the management server.  The remote file is
# relative to /tftpboot
#
atftp_put()
{
    _local_file=$1
    _remote_file=$2

    # Use this format if LOGGER=/usr/bin/logger
    #echo "put $_remote_file $local_file" | $ATFTP $TFTP_IP 2>&1 | $LOGGER

    # Use this format if LOGGER=write_log
    $LOGGER "`echo \"put $_local_file $_remote_file\" | $ATFTP $TFTP_IP 2>&1`"
} # END atftp_put


#
# Function to add a line to the status file on the management server.
# The status file is in the /csminstall/csm/status directory on the
# management server.
#
# The first argument to this function is the message.  It should be quoted
# if it contains spaces.  The second optional argument is a status.  This is
# typically used to send the exit code of a command to the status log.
#
# The format of the output message in the status file is:
```

```
#    <date>: <message>: <status>
# If <status> is not provided, the format of the message in the status file is:
#    <date>: <message>
#
# This function uses atftp to transfer the status file back and forth.  The
# atftp server should have been setup on the management server automatically
# when csm.server was installed.  The atftp client resides in the mounted
# /csminstall filesystem.  So this function cannot be used until after the
# /csminstall filesystem is mounted.
#
write_status()
{
    msg=$1;
    status=${2:-''}

    if [ "$status" ]
    then
 statusflag="-s $status"
    else
 statusflag=""
    fi

    /csminstall/csm/write_status -p $ATFTP -S $TFTP_IP -n $NODE_HOSTNAME $statusflag $msg

}  # END write_status


$LOGGER "Starting Install"

#netstat -rn | $LOGGER  # Use this format if LOGGER=/usr/bin/logger
$LOGGER "`netstat -rn`"  # Use this format if LOGGER=write_log

#
# Setup Syslog
#
# Turn syslog so we can use $LOGGER to send messages to syslogd running on a
# management node. Use $LOGGER.
#
# Uncomment this if you want all syslog messages from the node to be sent to
# the management server.  Be sure to also uncomment the section that says
# "Setup Syslog again" (below)
#
#mv /etc/syslog.conf /etc/syslog.conf.ORIG
#echo "*.* @$MGMTSVR_IP" >/etc/syslog.conf
#/etc/rc.d/init.d/syslog start
#$LOGGER "Install: syslog setup"

#
# Mount /csminstall
#
$LOGGER "Mounting $CSMINSTALL"
# Use this format if LOGGER=/usr/bin/logger
#mkdir $TFTPBOOT 2>&1 | $LOGGER
#mount -o nolock $NFS_IP:$TFTPBOOT $TFTPBOOT 2>&1 | $LOGGER
# Use this format if LOGGER=write_log
$LOGGER "`mkdir $CSMINSTALL 2>&1`"
$LOGGER "`mount -o nolock $NFS_IP:$CSMINSTALL $CSMINSTALL 2>&1`"

write_status "Kickstart RPM Installation Complete."
write_status "Starting Kickstart Post-Install."

# Run csmprereboot.  This adds csmfirstboot to the /etc/inittab file.
# It also mounts and unmounts /csminstall if needed.
/csminstall/csm/$CSM_VERSION/csmprereboot -m $MGMTSVR_HOSTNAME -n $NODE_HOSTNAME

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

```
#
# Hardcode eth0 (optional, recommended)
#
$LOGGER "Setting up eth0"
IP0=$(ifconfig eth0 | grep inet | awk '{print $2}' | awk -F: '{print $2}')
BC0=$(ifconfig eth0 | grep inet | awk '{print $3}' | awk -F: '{print $2}')
SM0=$(ifconfig eth0 | grep inet | awk '{print $4}' | awk -F: '{print $2}')
cd /etc/sysconfig/network-scripts
cp ifcfg-eth0 ifcfg-eth0.ORIG
echo "DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
IPADDR=$IP0
BROADCAST=$BC0
NETMASK=$SM0" >ifcfg-eth0

#
# Setup /etc/resolv.conf
#
$LOGGER "Setting up /etc/resolv.conf"
mv /etc/resolv.conf /etc/resolv.conf.ORIG
if [ "$DNSDOMAIN" ]
then
    echo "search $DNSDOMAIN" >/etc/resolv.conf
fi
for i in $(echo $NAMESERVERS | tr ',' ' ')
do
 echo "nameserver $i"
done >>/etc/resolv.conf

#
# Setup /etc/sysconfig/network
#
$LOGGER "Setting up /etc/sysconfig/network"
HOSTNAME=$(host $IP0 2>/dev/null | awk '{print $5}' | sed 's/\.$//')
SHORT_HOSTNAME=$(echo $HOSTNAME | awk -F. '{print $1}')
cp /etc/sysconfig/network /etc/sysconfig/network.ORIG
echo "NETWORKING=yes
HOSTNAME=$HOSTNAME
FORWARD_IPV4=yes
GATEWAYDEV=eth0
GATEWAY=$GATEWAY" >/etc/sysconfig/network

#
# Setup /etc/hosts
# Add myself and the management server
#
$LOGGER "Setting up /etc/hosts"
echo "127.0.0.1 localhost

$IP0 $HOSTNAME $SHORT_HOSTNAME
$MGMTSVR_IP $MGMTSVR_HOSTNAME $MGMTSVR_SHORTHOST" > /etc/hosts

#
# Add users (optional)
#
# The following line produces an encrypted password of "cluster"
# perl -e 'print crypt("cluster","Xa") . "\n";'p
#
$LOGGER "Adding users"
adduser admin -u 1000 -p `perl -e 'print crypt("cluster","Xa") . "\n";'`
#
# Enable rsh, add management server to /root/.rhosts and add 'rsh' to
# /etc/securetty to allow root to rsh to the node.
#
if [ "${SETUP_RSH}" = "Y" ]
```

```
then
 if [ -r /etc/xinetd.d/rsh ]
 then
  $LOGGER "Enable rsh"
  # Could use chkconfig to do this
  #perl -pi -e 's/^(\tdisable[^=]*=) yes/$1 no/' /etc/xinetd.d/rsh
  chkconfig rsh on

  $LOGGER "Allow root to rsh"
  echo "rsh" >> /etc/securetty

  $LOGGER "Setting up /root/.rhosts"
  cd /root
  echo "$MGMTSVR_SHORTHOST" >.rhosts
  echo "$MGMTSVR_HOSTNAME" >>.rhosts
  chmod 600 .rhosts
 fi
fi

#
# Setup PAM (optional)
#
# Uncomment this if you want to enable ftp, login, rlogin, rsh, sshd
#$LOGGER "Setup PAM"
#cd /etc/pam.d
#for i in ftp login rlogin rsh sshd
#do
# echo "account    required     /lib/security/pam_access.so" >>$i
#done
#echo "-:ALL EXCEPT root:ALL" >>/etc/security/access.conf
#cp /etc/security/access.conf /etc/security/access.conf.BOOT
#echo "cp /etc/security/access.conf.BOOT /etc/security/access.conf >/dev/null 2>&1" >>/etc/rc.d/rc.local


#
# Setup man paths in /etc/man.config.  Add CSM_MANPATH after /usr/local/man.
#
$LOGGER "Setup man paths"
cp -f /etc/man.config /etc/man.config.ORIG
FIXEDMANPATH=$(echo $CSM_MANPATH | sed 's/\//\\\//g')
perl -pi -e "s/^MANPATH[\t ]+\/usr\/local\/man$/MANPATH\t\/usr\/local\/man\nMANPATH $FIXEDMANPATH/" /etc/man.config

#
# Setup paths.
#  Add PATH definitions to /etc/profile.d/csm.sh and /etc/profile.d/csm.csh
#  Add section to /etc/csh.login that source's all the *.csh files
#     in /etc/profile.d/  (just like /etc/profile does for *.sh files).
#
$LOGGER "Setup PATHS"
echo "export PATH=\$PATH:$CSM_PATH" >/etc/profile.d/csm.sh
echo "setenv PATH \"\${PATH}:$CSM_PATH\"" >/etc/profile.d/csm.csh
chmod 755 /etc/profile.d/csm.*
echo "
test -d /etc/profile.d
if (\$status == 0) then
 set nonomatch
  foreach i ( /etc/profile.d/*.csh )
  test -f \$i
  if (\$status == 0) then
   source \$i
  endif
  end
 unset i nonomatch
endif" >>/etc/csh.login

#
# Setup services
```

```
#
$LOGGER "Setup services"
#chkconfig --del apmd
#chkconfig --del gpm
#chkconfig --del kudzu
#chkconfig --del lpd
#chkconfig --del pcmcia
#chkconfig --del linuxconf
#chkconfig --del sendmail
#chkconfig --del xfs
#chkconfig --del httpd
#chkconfig --del autofs
#chkconfig --del identd
#chkconfig --del isdn
#chkconfig --del pppoe
#chkconfig --level 345 ntpd on
#chkconfig --level 345 sshd on
#chkconfig --level 345 snmpd on
chkconfig telnet on

#
# Setup Syslog Again
#
# Uncomment this if you want all syslog messages from the node to be sent to
# the management server.  Be sure to also uncomment the section that says
# "Setup Syslog" (above)
#
#$LOGGER "Syslog setup again"
#echo "*.* @$MGMTSVR_IP" >/etc/syslog.conf

#
# Set local and remote complete install flag
# Put file in /dev/sda1
date >/boot/install_complete

####################### DO NOT ERASE THIS SECTION (begin) #####################
#
# Create a pxelinux configuration file for this node and tftp it to the
# management server.  This pxelinux file will cause the node to boot to
# the hard disk on the next reboot after the install is complete.  The
# CSM code will be installed during the first reboot of the hard disk.
#
$LOGGER "Update pxelinux configuration on management server"
HEX=$NODE_HEX

cd /tmp

#
# Use atftp to copy the HEX file over to the management server.
# The HEX file contains a new pxelinux configuration that causes the node
# to boot from hard disk on the next reboot.
#
echo "#boot" >$HEX
#$LOGGER "`echo \"get pxelinux.cfg/$HEX blah\" | $ATFTP $TFTP_IP 2>&1`"
atftp_get pxelinux.cfg/$HEX blah

grep SERIAL blah >>$HEX
echo "DEFAULT xCAT
LABEL xCAT
LOCALBOOT 0" >>$HEX

#$LOGGER "`echo \"put $HEX pxelinux.cfg/$HEX\" | $ATFTP $TFTP_IP 2>&1`"
atftp_put $HEX pxelinux.cfg/$HEX


#
# Copy the install log to the management server in /tftpboot/status
```

```
#
local_file=$KICKSTART_LOGFILE
remote_file="status/$NODE_HOSTNAME.install.log"
atftp_put $local_file $remote_file

#
# Tell the management server that the install is complete.
#
write_status "Kickstart Post-Install Complete."
write_status "Rebooting to hard disk."

#
# Unmount $CSMINSTALL
#
$LOGGER "Unmounting $CSMINSTALL"
cd /
umount $CSMINSTALL
rmdir $CSMINSTALL
####################### DO NOT ERASE THIS SECTION (end) #######################

cd /
$LOGGER "Install complete"
exit 0
```

RedHat 7.2 Kickstart configuration template file

The following is a sample Kickstart configuration template for RedHat 7.2. The template file is located in **/opt/csm/install/kscfg.tmpl.RedHat.7.2**. The most commonly updated sections of this template are shown in boldface type.

```
#
#Kickstart Configuration File for RedHat 7.2 Only
#

lang en_US
langsupport en_US
keyboard "us"

network --bootproto dhcp

#
# Where are the RedHat RPMs?
# nfs --server hostname.of.server --dir /path/to/RH/CD/image
#
# If your RedHat RPMs are not on the management server in the normal location,
# you can replace the NFS_HOSTNAME and NFS_DIR variables with your own
# values.
#
nfs --server #NFS_HOSTNAME# --dir #NFS_DIR#

#
# Clear the MBR
#
zerombr yes

#
# Wipe out the disk
#
clearpart --all

#
# Disk partition table. Customize it to fit your needs.
# /boot is strongly recommended
#
part /boot --size 50   --fstype ext3
part /     --size 1024 --fstype ext3
part swap  --size 1024 --ondisk sda
part /var  --size 1024 --fstype ext3
part /opt  --size 512  --fstype ext3
part /usr  --size 2048 --fstype ext3
#
# install or upgrade
#
install

#
# mouse selection
#
#mouse genericps/2 --emulthree
mouse none

#
# Select a time zone
# Add the --utc switch if your hardware clock is set to GMT
# Use the /usr/sbin/timeconfig tool to determine valid values.
#
# The timezone is set by default to the same timezone that the management
# server has.  This is found in /etc/sysconfig/clock on the management server.
#
timezone #TIMEZONE_UTC# #TIMEZONE#

#
# Don't set up X Windows.
#
skipx

#
# Perform the kickstart install in text mode
#
text

#
# Disable the firewall.  This allows NFS, dhcp, ssh, telnet and ftp to be used.
#
firewall --disabled

#
# To generate an encrypted root password use:
#
# perl -e 'print crypt("blah","Xa") . "\n";'p
# openssl passwd -apr1 -salt xxxxxxxx password
#
# where "blah" is your root password.
#
# --iscrypted is broken
#
#rootpw --iscrypted XaLGAVe1C41x2
#rootpw XaLGAVe1C41x2 --iscrypted
rootpw cluster
auth --useshadow --enablemd5

#
# Configure the Bootloader.  Use GRUB.
# If you want to add a GRUB password, add the --password option
#
bootloader  --location=mbr
#
# Reboot after installation
#
reboot

#
#end of section
```

```
#

#
# Choose the packages and package groups to install
# The following files must not be removed, or else csm will not work:
#   @ Network Support
#   pdksh
#   kernel-headers
#   rsh-server
#   nfs-utils
#   xinetd
#   tftp
#   binutils
#   others ????
#
#
%packages
@ Network Server
@ Utilities
@ X Window System
@ Legacy Application Support
kernel-smp
binutils
autofs
pdksh
tcsh
ntp
tftp
xinetd
rsh
rsh-server
telnet-server
nfs-utils
ucd-snmp
rsync
yp-tools
ypserv
ypbind
m4
sendmail-cf
%pre#
# Pre Install script
#
# This script may be modified as needed.  However, certain sections should
# not be altered or else the CSM installation will not work correctly. These
# sections are marked with the text "DO NOT ERASE THIS SECTION".
#

###################### DO NOT ERASE THIS SECTION (begin) #####################
#
# Set some variables first.  These get filled in by the management server.
export NODE_HOSTNAME=#NODE_HOSTNAME#
export NFS_IP=#NFS_IP#
export TFTP_IP=#TFTP_IP#
export CSM_VERSION=#CSM_VERSION#
export CSMINSTALL=/csminstall
export ATFTP=$CSMINSTALL/csm/$CSM_VERSION/atftp
export LOGFILE=/var/log/csm/install.log

write_log()
{
    mkdir -p /var/log/csm
    echo "`date`: $*" >> $LOGFILE
} # END write_log

LOGGER='write_log'
```

```
$LOGGER "`mkdir $CSMINSTALL 2>&1`"
$LOGGER "`mount -o nolock $NFS_IP:$CSMINSTALL $CSMINSTALL 2>1`"

# Update the status file on the management server to say that the install is
# starting.

msg="Reboot complete"
/csminstall/csm/write_status -v -p $ATFTP -S $TFTP_IP -n $NODE_HOSTNAME $msg

msg="Starting Kickstart Install"
/csminstall/csm/write_status -v -p $ATFTP -S $TFTP_IP -n $NODE_HOSTNAME $msg

umount $CSMINSTALL

####################### DO NOT ERASE THIS SECTION (end) #######################
exit 0

%post
#
# Post Install script
#
# This script may be modified as needed.  However, certain sections should
# not be altered or else the CSM installation will not work correctly. These
# sections are marked with the text "DO NOT ERASE THIS SECTION".
#

####################### DO NOT ERASE THIS SECTION (begin) #####################
#
# The LOGGER variable is used to send items to some kind of log.  It can be
# set to 'write_log' in order to call the 'write_log' function (defined below)
# which sends the log output to /var/log/csm/install.log on the node.
# If LOGGER is set to /usr/bin/logger, the Linux syslog facility is used
# instead.
#
#export LOGGER=/usr/bin/logger
export LOGGER=write_log

#
# Set variables
#
# The following variables are substituted into the kickstart template file
# from the setupks tool
#
# MGMTSVR_HOSTNAME   = management server (Long Hostname)
# MGMTSVR_SHORTHOST  = management server (Short Hostname)
# MGMTSVR_IP         = management server (IP)
# NODE_HOSTNAME      = node hostname (Long Hostname)
# NODE_HEX           = node IP address in hexidecimal format
# NFS_HOSTNAME       = NFS Server (Long Hostname) - same as MGMTSVR_HOSTNAME
# NFS_IP             = NFS Server (IP) - same as MGMTSVR_IP
# NFS_DIR            = Path to RedHat CD image on NFS Server
# TFTP_IP            = TFTP Server (IP) - same as MGMTSVR_IP
# NAMESERVERS        = list of comma delimited name servers(IPs or Hostnames)
# DNSDOMAIN          = DNS domain name of the node(NAME)
# SETUP_RSH          = Set to "Y" if $DSH_REMOTE_CMD variable on the
#                      management server is set to "rsh" or is blank.
# CSM_VERSION        = The version of CSM that is being installed
# TIMEZONE           = Timezone string of the Management Server
# TIMEZONE_UTC       = Timezone Universal System Time Setting
#                      on the Management Server (either --utc or blank)
#
export KICKSTART_LOGFILE=/var/log/csm/install.log
export MGMTSVR_HOSTNAME=#MGMTSVR_HOSTNAME#
export MGMTSVR_SHORTHOST=#MGMTSVR_SHORTHOST#
export MGMTSVR_IP=#MGMTSVR_IP#
export NODE_HOSTNAME=#NODE_HOSTNAME#
export NODE_HEX=#NODE_HEX#
export NFS_HOSTNAME=#NFS_HOSTNAME#
export NFS_IP=#NFS_IP#
export TFTP_IP=#TFTP_IP#
export NAMESERVERS=#NAMESERVERS#
export DNSDOMAIN=#DNSDOMAIN#
export SETUP_RSH=#SETUP_RSH#
export CSM_VERSION=#CSM_VERSION#

#
# Hardcode some other variables
#
export GATEWAY=$(netstat -rn | grep 0.0.0.0 | tail -1 | awk '{print $2}')
export CSM_MANPATH=/opt/man
export CSM_PATH=/opt/bin
export TFTPBOOT=/tftpboot
export CSMINSTALL=/csminstall
export ATFTP=$CSMINSTALL/csm/$CSM_VERSION/atftp

#
# Function to add some text to the install.log file on the node.
#
write_log()
{
    mkdir -p /var/log/csm
    echo "`date`: $*" >> $KICKSTART_LOGFILE
} # END write_log

#
# Use atftp to get a file from the management server.  The remote file is
# relative to /tftpboot
#
atftp_get()
{
    _remote_file=$1
    _local_file=$2

    # Use this format if LOGGER=/usr/bin/logger
    #echo "get $_local_file $remote_file" | $ATFTP $TFTP_IP 2>&1 | $LOGGER

    # Use this format if LOGGER=write_log
    $LOGGER "`echo \"get $_remote_file $_local_file\" | $ATFTP $TFTP_IP 2>&1`"
} # END atftp_get

#
# Use atftp to put a file to the management server.  The remote file is
# relative to /tftpboot
#
atftp_put()
{
    _local_file=$1
    _remote_file=$2

    # Use this format if LOGGER=/usr/bin/logger
    #echo "put $_remote_file $local_file" | $ATFTP $TFTP_IP 2>’1 | $LOGGER

    # Use this format if LOGGER=write_log
    $LOGGER "`echo \"put $_local_file $_remote_file\" | $ATFTP $TFTP_IP 2>&1`"
} # END atftp_put


#
# Function to add a line to the status file on the management server.
# The status file is in the /csminstall/csm/status directory on the
# management server.
#
# The first argument to this function is the message.  It should be quoted
# if it contains spaces.  The second optional argument is a status.  This is
# typically used to send the exit code of a command to the status log.
#
# The format of the output message in the status file is:
#    <date>: <message>: <status>
# If <status> is not provided, the format of the message in the status file is:
#    <date>: <message>
#
# This function uses atftp to transfer the status file back and forth.  The
# atftp server should have been setup on the management server automatically
# when csm.server was installed.  The atftp client resides in the mounted
# /csminstall filesystem.  So this function cannot be used until after the
# /csminstall filesystem is mounted.
#
write_status()
{
    msg=$1;
    status=${2:-''}

    if [ "$status" ]
    then
 statusflag="-s $status"
    else
 statusflag=""
    fi

    /csminstall/csm/write_status -p $ATFTP -S $TFTP_IP -n $NODE_HOSTNAME $statusflag $msg

}  # END write_status


$LOGGER "Starting Install"

#netstat -rn | $LOGGER  # Use this format if LOGGER=/usr/bin/logger
$LOGGER "`netstat -rn`"  # Use this format if LOGGER=write_log

#
# Setup Syslog
#
# Turn syslog so we can use $LOGGER to send messages to syslogd running on a
# management node. Use $LOGGER.
#
# Uncomment this if you want all syslog messages from the node to be sent to
# the management server.  Be sure to also uncomment the section that says
# "Setup Syslog again" (below)
#
#mv /etc/syslog.conf /etc/syslog.conf.ORIG
#echo "*.* @$MGMTSVR_IP" >/etc/syslog.conf
#/etc/rc.d/init.d/syslog start
#$LOGGER "Install: syslog setup"

#
# Mount /csminstall
#
$LOGGER "Mounting $CSMINSTALL"
# Use this format if LOGGER=/usr/bin/logger
#mkdir $TFTPBOOT 2>&1 | $LOGGER
#mount -o nolock $NFS_IP:$TFTPBOOT $TFTPBOOT 2>&1 | $LOGGER
# Use this format if LOGGER=write_log
$LOGGER "`mkdir $CSMINSTALL 2>&1`"
$LOGGER "`mount -o nolock $NFS_IP:$CSMINSTALL $CSMINSTALL 2>&1`"

write_status "Kickstart RPM Installation Complete."
write_status "Starting Kickstart Post-Install."

# Run csmprereboot.  This adds csmfirstboot to the /etc/inittab file.
# It also mounts and unmounts /csminstall if needed.
/csminstall/csm/$CSM_VERSION/csmprereboot -m $MGMTSVR_HOSTNAME -n $NODE_HOSTNAME
```

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

#
# Hardcode eth0 (optional, recommended)
#
$LOGGER "Setting up eth0"
IP0=$(ifconfig eth0 | grep inet | awk '{print $2}' | awk -F: '{print $2}')
BC0=$(ifconfig eth0 | grep inet | awk '{print $3}' | awk -F: '{print $2}')
SM0=$(ifconfig eth0 | grep inet | awk '{print $4}' | awk -F: '{print $2}')
cd /etc/sysconfig/network-scripts
cp ifcfg-eth0 ifcfg-eth0.ORIG
echo "DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
IPADDR=$IP0
BROADCAST=$BC0
NETMASK=$SM0" >ifcfg-eth0

#
# Setup /etc/resolv.conf
#
$LOGGER "Setting up /etc/resolv.conf"
mv /etc/resolv.conf /etc/resolv.conf.ORIG
if [ "$DNSDOMAIN" ]
then
    echo "search $DNSDOMAIN" >/etc/resolv.conf
fi
for i in $(echo $NAMESERVERS | tr ',' ' ')
do
 echo "nameserver $i"
done >>/etc/resolv.conf

#
# Setup /etc/sysconfig/network
#
$LOGGER "Setting up /etc/sysconfig/network"
HOSTNAME=$(host $IP0 2>/dev/null | awk '{print $5}' | sed 's/\.$//')
SHORT_HOSTNAME=$(echo $HOSTNAME | awk -F. '{print $1}')
cp /etc/sysconfig/network /etc/sysconfig/network.ORIG
echo "NETWORKING=yes
HOSTNAME=$HOSTNAME
FORWARD_IPV4=yes
GATEWAYDEV=eth0
GATEWAY=$GATEWAY" >/etc/sysconfig/network

#
# Setup /etc/hosts
# Add myself and the management server
#
$LOGGER "Setting up /etc/hosts"
echo "127.0.0.1 localhost

$IP0 $HOSTNAME $SHORT_HOSTNAME
$MGMTSVR_IP $MGMTSVR_HOSTNAME $MGMTSVR_SHORTHOST" > /etc/hosts

#
# Add users (optional)
#
# The following line produces an encrypted password of "cluster"
# perl -e 'print crypt("cluster","Xa") . "\n";'p
#
$LOGGER "Adding users"
adduser admin -u 1000 -p `perl -e 'print crypt("cluster","Xa") . "\n";'`
#
# Enable rsh, add management server to /root/.rhosts and add 'rsh' to
# /etc/securetty to allow root to rsh to the node.
#
if [ "${SETUP_RSH}" = "Y" ]
then
 if [ -r /etc/xinetd.d/rsh ]
 then
  $LOGGER "Enable rsh"
  # Could use chkconfig to do this
  #perl -pi -e 's/^(\tdisable[^=]*=) yes/$1 no/' /etc/xinetd.d/rsh
  chkconfig rsh on

  $LOGGER "Allow root to rsh"
  echo "rsh" >> /etc/securetty

  $LOGGER "Setting up /root/.rhosts"
  cd /root
  echo "$MGMTSVR_SHORTHOST" >.rhosts
  echo "$MGMTSVR_HOSTNAME" >>.rhosts
  chmod 600 .rhosts
 fi
fi

#
# Setup PAM (optional)
#
# Uncomment this if you want to enable ftp, login, rlogin, rsh, sshd
#$LOGGER "Setup PAM"
#cd /etc/pam.d
#for i in ftp login rlogin rsh sshd
#do
# echo "account    required     /lib/security/pam_access.so" >>$i
#done
#echo "-:ALL EXCEPT root:ALL" >>/etc/security/access.conf
#cp /etc/security/access.conf /etc/security/access.conf.BOOT
#echo "cp /etc/security/access.conf.BOOT /etc/security/access.conf >/dev/null 2>&1" >>/etc/rc.d/rc.local


#
# Setup man paths in /etc/man.config.  Add CSM_MANPATH after /usr/local/man.
#
$LOGGER "Setup man paths"
cp -f /etc/man.config /etc/man.config.ORIG
FIXEDMANPATH=$(echo $CSM_MANPATH | sed 's/\//\\\//g')
perl -pi -e "s/^MANPATH[\t ]+\/usr\/local\/man$/MANPATH\t\/usr\/local\/man\nMANPATH $FIXEDMANPATH/" /etc/man.config

#
# Setup paths.
#  Add PATH definitions to /etc/profile.d/csm.sh and /etc/profile.d/csm.csh
#  Add section to /etc/csh.login that source's all the *.csh files
#     in /etc/profile.d/  (just like /etc/profile does for *.sh files).
#
$LOGGER "Setup PATHS"
echo "export PATH=\$PATH:$CSM_PATH" >/etc/profile.d/csm.sh
echo "setenv PATH \"\${PATH}:$CSM_PATH\"" >/etc/profile.d/csm.csh
chmod 755 /etc/profile.d/csm.*
echo "
test -d /etc/profile.d
if (\$status == 0) then
 set nonomatch
  foreach i ( /etc/profile.d/*.csh )
  test -f \$i
  if (\$status == 0) then
   source \$i
  endif
  end
 unset i nonomatch
endif" >>/etc/csh.login
```

```
#
# Setup services
#
$LOGGER "Setup services"
#chkconfig --del apmd
#chkconfig --del gpm
#chkconfig --del kudzu
#chkconfig --del lpd
#chkconfig --del pcmcia
#chkconfig --del linuxconf
#chkconfig --del sendmail
#chkconfig --del xfs
#chkconfig --del httpd
#chkconfig --del autofs
#chkconfig --del identd
#chkconfig --del isdn
#chkconfig --del pppoe
#chkconfig --level 345 ntpd on
#chkconfig --level 345 sshd on
#chkconfig --level 345 snmpd on
chkconfig telnet on

#
# Setup Syslog Again
#
# Uncomment this if you want all syslog messages from the node to be sent to
# the management server.  Be sure to also uncomment the section that says
# "Setup Syslog" (above)
#
#$LOGGER "Syslog setup again"
#echo "*.* @$MGMTSVR_IP" >/etc/syslog.conf

#
# Set local and remote complete install flag
# Put file in /dev/sda1
date >/boot/install_complete

###################### DO NOT ERASE THIS SECTION (begin) #####################
#
# Create a pxelinux configuration file for this node and tftp it to the
# management server.  This pxelinux file will cause the node to boot to
# the hard disk on the next reboot after the install is complete.  The
# CSM code will be installed during the first reboot of the hard disk.
#
$LOGGER "Update pxelinux configuration on management server"
HEX=$NODE_HEX

cd /tmp

#
# Use atftp to copy the HEX file over to the management server.
# The HEX file contains a new pxelinux configuration that causes the node
# to boot from hard disk on the next reboot.
#
echo "#boot" >$HEX
#$LOGGER "`echo \"get pxelinux.cfg/$HEX blah\" | $ATFTP $TFTP_IP 2>&1`"
atftp_get pxelinux.cfg/$HEX blah

grep SERIAL blah >>$HEX
echo "DEFAULT xCAT
LABEL xCAT
LOCALBOOT 0" >>$HEX

#$LOGGER "`echo \"put $HEX pxelinux.cfg/$HEX\" | $ATFTP $TFTP_IP 2>&1`"
atftp_put $HEX pxelinux.cfg/$HEX
```

```
#
# Copy the install log to the management server in /tftpboot/status
#
local_file=$KICKSTART_LOGFILE
remote_file="status/$NODE_HOSTNAME.install.log"
atftp_put $local_file $remote_file

#
# Tell the management server that the install is complete.
#
write_status "Kickstart Post-Install Complete."
write_status "Rebooting to hard disk."

#
# Unmount $CSMINSTALL
#
$LOGGER "Unmounting $CSMINSTALL"
cd /
umount $CSMINSTALL
rmdir $CSMINSTALL
####################### DO NOT ERASE THIS SECTION (end) #######################

cd /
$LOGGER "Install complete"
exit 0
```

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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

Poughkeepsie, New York 12601-5400
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following names are trademarks or registered trademarks in the United States, other countries, or both:

- IBM, AIX, AIX 5L, Netfinity, RS/6000, SP, xSeries, eServer, and the @server logo are trademarks or registered trademarks of International Business Machines Corporation.
- CPS is a trademark of Avocent Corporation.
- Equinox is a trademark of Equinox Systems, Inc.
- IntelliServer is a registered trademark of Computone Corporation.
- 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 are 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 uses or can be configured to use the following 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**.

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.

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.

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 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 *CSM for Linux: Software 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.

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 Advanced TFTP (atftp) 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.

Any software identified herein as being licensed under the GPL should not be construed as being offered herein under any license other than the GPL.

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

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®
Software Planning and Installation Guide
Version 1.2

Publication No. SA22-7853-03

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
SA22-7853-03

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie NY 12601-5400

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Program Number: 5765–E88

SA22-7853-03