High Availability Cluster
Multi-Processing for AIX

Troubleshooting Guide

Version 5.4.1

SC23-5177-04

Fifth Edition (October 2007)

Before using the information in this book, read the general information in Notices for HACMP Troubleshooting Guide.

This edition applies to HACMP for AIX, version 5.4.1 and to all subsequent releases of this product until otherwise indicated in new editions.

Contents

Contents

About This Guide

This guide provides information necessary to troubleshoot the High Availability Cluster Multi-Processing for AIX v.5.4.1 software. For information on planning an HACMP™ cluster, see the *Planning Guide.* For information on installation, see the *Installation Guide*. For information about configuring and managing an HACMP cluster, see the *Administration Guide*.

The following table provides version and manual part numbers for the *Troubleshooting Guide*.

HACMP Version	Book Name	Book Number
5.4.1	*Troubleshooting Guide*	SC23-5177-04
5.4	*Troubleshooting Guide*	SC23-5177-03
5.3 last update 7/2006	*Troubleshooting Guide*	SC23-5177-02
5.3 update 8/2005	*Troubleshooting Guide*	SC23-5177-01
5.3	*Troubleshooting Guide*	SC23-5177-00
5.2 last update 10/2005	*Administration and Troubleshooting Guide*	SC23-4862-05

Who Should Use This Guide

This guide is intended for system administrators and customer engineers responsible for configuring, managing, and troubleshooting an HACMP cluster. As a prerequisite for maintaining the HACMP software, you should be familiar with:

- IBM System p™ system components (including disk devices, cabling, and network adapters)
- The AIX operating system, including the Logical Volume Manager subsystem
- The System Management Interface Tool (SMIT)
- Communications, including the TCP/IP subsystem.

Highlighting

This guide uses the following highlighting conventions:

Italic	Identifies new terms or concepts, or indicates emphasis.
Bold	Identifies routines, commands, keywords, files, directories, menu items, and other items whose actual names are predefined by the system.
`Monospace`	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of program code similar to what you might write as a programmer, messages from the system, or information that you should actually type.

ISO 9000

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

HACMP Publications

The HACMP software comes with the following publications:

- *HACMP for AIX Release Notes* in **/usr/es/sbin/cluster/release_notes** describe issues relevant to HACMP on the AIX platform: latest hardware and software requirements, last-minute information on installation, product usage, and known issues.
- *HACMP on Linux Release Notes* in **/usr/es/sbin/cluster/release_notes.linux/** describe issues relevant to HACMP on the Linux platform: latest hardware and software requirements, last-minute information on installation, product usage, and known issues.
- *HACMP for AIX: Administration Guide,* SC23-4862
- *HACMP for AIX: Concepts and Facilities Guide,* SC23-4864
- *HACMP for AIX: Installation Guide,* SC23-5209
- *HACMP for AIX: Master Glossary,* SC23-4867
- *HACMP for AIX: Planning Guide,* SC23-4861
- *HACMP for AIX: Programming Client Applications*, SC23-4865
- *HACMP for AIX: Troubleshooting Guide,* SC23-5177
- *HACMP on Linux: Installation and Administration Guide,* SC23-5211
- *HACMP for AIX: Smart Assist Developer's Guide,* SC23-5210
- *IBM International Program License Agreement.*

HACMP/XD Publications

The HACMP Extended Distance (HACMP/XD) software solutions for disaster recovery, added to the base HACMP software, enable a cluster to operate over extended distances at two sites. HACMP/XD publications include the following:

- *HACMP/XD for Geographic LVM (GLVM): Planning and Administration Guide,* SA23-1338
- *HACMP/XD for HAGEO Technology: Concepts and Facilities Guide, SC23-1922*
- *HACMP/XD for HAGEO Technology: Planning and Administration Guide,* SC23-1886
- *HACMP/XD for Metro Mirror: Planning and Administration Guide,* SC23-4863.

HACMP Smart Assist Publications

The HACMP Smart Assist software helps you quickly add an instance of certain applications to your HACMP configuration so that HACMP can manage their availability. The HACMP Smart Assist publications include the following:

- *HACMP Smart Assist for DB2 User's Guide, SC23-5179*
- *HACMP Smart Assist for Oracle User's Guide, SC23-5178*

- *HACMP Smart Assist for WebSphere User's Guide, SC23-4877*
- *HACMP for AIX 5L: Smart Assist Developer's Guide, SC23-5210*
- *HACMP Smart Assist Release Notes.*

IBM AIX Publications

The following publications offer more information about IBM technology related to or used by HACMP:

- *RS/6000 SP High Availability Infrastructure*, SG24-4838
- *IBM AIX v.5.3 Security Guide,* SC23-4907
- *IBM Reliable Scalable Cluster Technology for AIX and Linux: Group Services Programming Guide and Reference,* SA22-7888
- *IBM Reliable Scalable Cluster Technology for AIX and Linux: Administration Guide,* SA22-7889
- *IBM Reliable Scalable Cluster Technology for AIX: Technical Reference*, SA22-7890
- *IBM Reliable Scalable Cluster Technology for AIX: Messages*, GA22-7891.

Accessing Publications

Use the following Internet URLs to access online libraries of documentation:

AIX, IBM eServer pSeries, and related products:

http://www.ibm.com/servers/aix/library

AIX v.5.3 publications:

http://www.ibm.com/servers/eserver/pseries/library/

WebSphere Application Server publications:

Search the IBM website to access the WebSphere Application Server Library

DB2 Universal Database Enterprise Server Edition publications:

http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v8pubs.d2w/en_main#V8PDF

Tivoli Directory Server publications:

http://publib.boulder.ibm.com/tividd/td/IBMDirectoryServer5.1.html

IBM Welcomes Your Comments

You can send any comments via e-mail to hafeedbk@us.ibm.com. 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.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States or other countries:

- AFS
- AIX
- DFS
- @server
- eServer Cluster 1600
- Enterprise Storage Server
- HACMP
- IBM
- NetView
- RS/6000
- Scalable POWERParallel Systems
- Series p
- Series x
- Shark
- SP
- WebSphere
- Red Hat Enterprise Linux (RHEL)
- SUSE Linux Enterprise Server
- RPM Package Manager for Linux and other Linux trademarks.

UNIX is a registered trademark in the United States and other countries and is licensed exclusively through The Open Group.

Linux is a registered trademark in the United States and other countries and is licensed exclusively through the GNU General Public License.

Other company, product, and service names may be trademarks or service marks of others.

Chapter 1: Troubleshooting HACMP Clusters

This chapter presents the recommended troubleshooting strategy for an HACMP cluster. It describes the problem determination tools available from the HACMP main SMIT menu. This guide also includes information on tuning the cluster for best performance, which can help you avoid some common problems.

For details on how to use the various log files to troubleshoot the cluster see Chapter 2: Using Cluster Log Files. For hints on how to check system components if using the log files does *not* help with the problem, and a list of solutions to common problems that may occur in an HACMP environment see Chapter 3: Investigating System Components and Solving Common Problems.

For information specific to RSCT daemons and diagnosing RSCT problems, see the following IBM publications:

- *IBM Reliable Scalable Cluster Technology for AIX and Linux: Group Services Programming Guide and Reference,* SA22-7888
- *IBM Reliable Scalable Cluster Technology for AIX and Linux: Administration Guide,* SA22-7889
- *IBM Reliable Scalable Cluster Technology for AIX: Technical Reference*, SA22-7890
- *IBM Reliable Scalable Cluster Technology for AIX: Messages*, GA22-7891.

Note: This chapter presents the default locations of log files. If you redirected any logs, check the appropriate location. For additional information, see Chapter 2: Using Cluster Log Files.

The main sections of this chapter include:

- Troubleshooting an HACMP Cluster Overview
- Using the Problem Determination Tools
- Configuring Cluster Performance Tuning
- Resetting HACMP Tunable Values
- Sample Custom Scripts.

Troubleshooting an HACMP Cluster Overview

Typically, a functioning HACMP cluster requires minimal intervention. If a problem does occur, diagnostic and recovery skills are essential. Therefore, troubleshooting requires that you identify the problem quickly and apply your understanding of the HACMP software to restore the cluster to full operation. In general, troubleshooting an HACMP cluster involves:

- Becoming aware that a problem exists
- Determining the source of the problem
- Correcting the problem.

Becoming Aware of the Problem

When a problem occurs within an HACMP cluster, you will most often be made aware of it through:

- End user complaints, because they are *not* able to access an application running on a cluster node
- One or more error messages displayed on the system console or in another monitoring program.

There are other ways you can be notified of a cluster problem, through mail notification, or pager notification and text messaging:

- *Mail Notification.* Although HACMP standard components do *not* send mail to the system administrator when a problem occurs, you can create a mail notification method as a pre- or post-event to run before or after an event script executes. In an HACMP cluster environment, mail notification is effective and highly recommended. See the *Planning Guide* for more information.
- *Remote Notification.* You can also define a notification method—numeric or alphanumeric page, or an text messaging notification to any address including a cell phone—through the SMIT interface to issue a customized response to a cluster event. For more information, see the chapter on customizing cluster events in the *Planning Guide*.
  - *Pager Notification.* You can send messages t*o* a pager number on a given event. You can send textual information to pagers that support text display (alphanumeric page), and numerical messages to pagers that only display numbers.
  - *Text Messaging*. You can send cell phone text messages using a standard data modem and telephone land line through the standard Telocator Alphanumeric Protocol (TAP)—your provider must support this service.

    You can also issue a text message using a Falcom-compatible GSM modem to transmit SMS (Short Message Service) text-message notifications wirelessly. SMS messaging requires an account with an SMS service provider. GSM modems take TAP modem protocol as input through a RS232 line or USB line, and send the message wirelessly to the providers' cell phone tower. The provider forwards the message to the addressed cell phone. Each provider has a Short Message Service Center (SMSC).

  For each person, define remote notification methods that contain all the events and nodes so you can switch the notification methods as a unit when responders change.

Note: Manually distribute each message file to each node. HACMP does *not* automatically distribute the file to other nodes during synchronization unless the File Collections utility is set up specifically to do so. See the Managing HACMP File Collections section in Chapter 7: Verifying and Synchronizing a Cluster Configuration of the *Administration Guide*.

Messages Displayed on System Console

The HACMP system generates descriptive messages when the scripts it executes (in response to cluster events) start, stop, or encounter error conditions. In addition, the daemons that make up an HACMP cluster generate messages when they start, stop, encounter error conditions, or change state. The HACMP system writes these messages to the system console and to one or more cluster log files. Errors may also be logged to associated system files, such as the **snmpd.log** file.

For information on all of the cluster log files see Chapter 2: Using Cluster Log Files.

Determining a Problem Source

If a problem with HACMP has been detected, perform the following actions for initial problem analysis:

1. Collect an HACMP snapshot with the **snap -e** command. This should be done as soon as possible after the problem has been detected because the collected log files contain a time window of error.
2. Establish the state of the cluster and resource groups using the **/usr/es/sbin/cluster/clstat**, and **/usr/es/sbin/cluster/utilities/clRGinfo** commands.
3. If an event error occurred, inspect the **/var/hacmp/log/hacmp.out** file to locate the error. If an AIX command failed, proactively collect further debug data for the corresponding AIX component, using the **snap** command. The most commonly requested flag for further problem determination for HACMP is **snap -gGtL**.
4. Consult the /**var/hacmp/log/clverify.log**, and **/var/hacmp/log/autoverify.log** files for the result of the most recent cluster verification. Run cluster verification.
5. If a C-SPOC command failed, consult the **/var/hacmp/log/cspoc.log.long** file.
6. Verify network connectivity between nodes.
7. Inspect the error log (**errpt -a**) to establish if errors have been logged in the time window of failure.

Stopping the Cluster Manager

To fix some cluster problems, you must stop the Cluster Manager on the failed node and have a surviving node take over its shared resources. If the cluster is in reconfiguration, it can only be brought down by stopping it and placing the resource group in an UNMANAGED state. The surviving nodes in the cluster will interpret this kind of stop as a **node_down** event, but will *not* attempt to take over resources. The resources will still be available on that node. You can then begin the troubleshooting procedure.

If all else fails, stop the HACMP cluster services on all cluster nodes. Then, manually start the application that the HACMP cluster event scripts were attempting to start and run the application without the HACMP software. This may require varying on volume groups, mounting file systems, and enabling IP addresses. With the HACMP cluster services stopped on all cluster nodes, correct the conditions that caused the initial problem.

Using the AIX Data Collection Utility

Use the AIX **snap** command to collect data from an HACMP cluster.

Flag **-e** collects data that aids IBM support in troubleshooting a problem with HACMP and its interaction with other components. In particular, flag **-e** collects all log files of HACMP utilities, ODMs maintained by HACMP, some AIX ODMs, and AIX configuration data most commonly required (such as LVM, TCP/IP and installp information). The **snap -e** command runs **/usr/sbin/rsct/bin/phoenix.snap**, which collects data of the Group Services and Topology Services RSCT subsystems.

The HACMP snapshot should be collected as soon as possible after a problem has been encountered with HACMP, to ensure that the data pertaining to the time window of error are contained in the log files.

The **snap -e** command relies on the Cluster Communication Daemon subsystem (**clcomdES**), to collect data. If this subsystem is affected by an error, the **snap -e** command might fail. In this case, collect the following data on all cluster nodes:

- tar archive of directory **/var/hacmp**
- **/usr/sbin/rsct/bin/phoenix.snap**
- tar archives of directories **/etc/es/objrepos** and **/usr/es/sbin/cluster/etc/objrepos/active**
- **snap -cgGLt**

For more information on the **snap** command, see the *AIX Version 6.1 Commands Reference, Volume 5*.

Checking a Cluster Configuration with Online Planning Worksheets

The Online Planning Worksheets application lets you view a cluster definition for the following:

- Local HACMP cluster running HACMP 5.2 or greater
- Cluster worksheets file created from SMIT or from Online Planning Worksheets.

You can use a worksheets file to view information for a cluster configuration and to troubleshoot cluster problems. The Online Planning Worksheets application lets you review definition details on the screen in an easy-to-read format and lets you create a printable formatted report.

WARNING: Although you can import a cluster definition and save it, some of the data is informational only. Making changes to informational components does *not* change the actual configuration on the system if the worksheets file is exported. For information about informational components in a worksheets file, see the section Entering Data in Chapter 9: Using Online Planning Worksheets in the *Planning Guide*.

Note: Cluster definition files and their manipulation in the Online Planning Worksheets application supplement, but do *not* replace cluster snapshots.

For more information about using cluster definition files in the Online Planning Worksheets application, see Chapter 9: Using Online Planning Worksheets in the *Planning Guide*.

Using HACMP Diagnostic Utilities

Both HACMP and AIX supply many diagnostic tools. The key HACMP diagnostic tools (in addition to the cluster logs and messages) include:

- **clRGinfo** provides information about resource groups and for troubleshooting purposes. For more information see Chapter 10: Monitoring an HACMP Cluster in the *Administration Guide*.
- **clstat** reports the status of key cluster components—the cluster itself, the nodes in the cluster, the network interfaces connected to the nodes, the service labels, and the resource groups on each node. For more information see Chapter 10: Monitoring an HACMP Cluster in the *Administration Guide*.
- **clsnapshot** allows you to save in a file a record of all the data that defines a particular cluster configuration. For more information see the section Using the Cluster Snapshot Utility to Check Cluster Configuration. and Creating (Adding) a Cluster Snapshot section in Chapter 18: Saving and Restoring Cluster Configurations in the *Administration Guide*.
- **cldisp** utility displays resource groups and their startup, fallover, and fallback policies. For more information Chapter 10: Monitoring an HACMP Cluster in the *Administration Guide*.
- SMIT **Problem Determination Tools**, for information see the section Using the Problem Determination Tools in this chapter.

Using the Cluster Snapshot Utility to Check Cluster Configuration

The HACMP cluster snapshot facility (**/usr/es/sbin/cluster/utilities/clsnapshot**) allows you to save in a file, a record of all the data that defines a particular cluster configuration. You can use this *snapshot* for troubleshooting cluster problems.

The cluster snapshot saves the data stored in the HACMP Configuration Database classes. In addition to this Configuration Database data, a cluster snapshot also includes output generated by various HACMP and standard AIX commands and utilities. This data includes the current state of the cluster, node, network, and network interfaces as viewed by each cluster node, and the state of any running HACMP daemons. It may also include additional user-defined information if there are custom snapshot methods in place.

In HACMP 5.1 and up, by default, HACMP no longer collects the cluster log files when you create the cluster snapshot. You can still specify in SMIT that the logs be collected if you want them. Skipping the logs collection reduces the size of the snapshot and reduces the running time of the snapshot utility.

For more information on using the cluster snapshot utility, see Chapter 18: Saving and Restoring Cluster Configurations in the *Administration Guide*.

Working with SMIT Problem Determination Tools

The SMIT **Problem Determination Tools** menu includes the options offered by cluster snapshot utility, to help you diagnose and solve problems. For more information see the following section on Using the Problem Determination Tools in this chapter.

Verifying Expected Behavior

When the highly available applications are up and running, verify that end users can access the applications. If *not*, you may need to look elsewhere to identify problems affecting your cluster. The remaining chapters in this guide describe ways in which you should be able to locate potential problems.

Using the Problem Determination Tools

The **Problem Determination Tools** menu options are described in the following sections:

- HACMP Verification
- Viewing Current State
- HACMP Log Viewing and Management
- Recovering from HACMP Script Failure
- Restoring HACMP Configuration Database from an Active Configuration
- Release Locks Set by Dynamic Reconfiguration
- Clear SSA Disk Fence Registers
- HACMP Cluster Test Tool
- HACMP Trace Facility
- HACMP Event Emulation
- HACMP Error Notification
- Opening a SMIT Session on a Node.

HACMP Verification

Select this option from the **Problem Determination Tools** menu to verify that the configuration on all nodes is synchronized, set up a custom verification method, or set up automatic cluster verification.

Verify HACMP Configuration	Select this option to verify cluster topology resources.
Configure Custom Verification Method	Use this option to add, show and remove custom verification methods.
Automatic Cluster Configuration Monitoring	Select this option to automatically verify the cluster every twenty-four hours and report results throughout the cluster.

Verify HACMP Configuration

To verify cluster topology resources and custom-defined verification methods:

1. Enter `smit hacmp`
2. In SMIT, select **Problem Determination Tools > HACMP Verification > Verify HACMP Configuration**.
3. Enter field values as follows:

HACMP Verification Method	By default, **Pre-Installed** will run all verification methods shipped with HACMP and HACMP/XD verification (if applicable or user-provided). You can select this field to run all **Pre-Installed** programs or select **none** to specify a previously defined custom verification method.
Custom Defined Verification Method	Enter the name of a custom defined verification method. Press F4 for a list of previously defined verification methods. By default, when no methods are selected, and **none** is selected in the **Base HACMP Verification Method** field, verify and synchronize will *not* check the base verification methods, and will generate an error message. The order in which verification methods are listed determines the sequence in which the methods run. This sequence remains the same for subsequent verifications until different methods are selected. Selecting **All** verifies all custom-defined methods. See Adding a Custom Verification Method in Chapter 7: Verifying and Synchronizing a Cluster Configuration in the *Administration Guide* for information on adding or viewing a customized verification method.
Error Count	By default, **Verify HACMP Configuration** will continue to run after encountering an error in order to generate a full list of errors. To cancel the program after a specific number of errors, type the number in this field.

Log File to store output	Enter the name of an output file in which to store verification output. By default, verification output is also stored in the **/usr/es/sbin/cluster/wsm/logs/ wsm_smit.log** file.
Verify Changes Only?	Select **no** to run all verification checks that apply to the current cluster configuration. Select **yes** to run only the checks related to parts of the HACMP configuration that have changed. The **yes** mode has no effect on an inactive cluster. **Note:** The **yes** option only relates to cluster Configuration Databases. If you have made changes to the AIX configuration on your cluster nodes, you should select **no**. Only select **yes** if you have made no changes to the AIX configuration.
Logging	Selecting **on** displays all output to the console that normally goes to the **/var/hacmp/clverify/ clverify.log**. The default is **off**.

Configure Custom Verification Method

You may want to add a custom verification method to check for a particular issue on your cluster. For example, you could add a script to check for the version of an application. You could include an error message to display and to write to the **clverify.log** file.

For information on adding or viewing a customized verification method, see the Adding a Custom Verification Method section in Chapter 7: Verifying and Synchronizing a Cluster Configuration in the *Administration Guide*.

Automatic Monitoring and Verification of Cluster Configuration

The cluster verification utility runs on one user-selectable HACMP cluster node once every 24 hours. By default, the first node in alphabetical order runs the verification at midnight. During verification, any errors that might cause problems at some point in the future are displayed. You can change the defaults, by selecting a node and time that suit your configuration.

If the selected node is unavailable (powered off), verification does *not* run the automatic monitoring. When cluster verification completes on the selected cluster node, this node notifies the other cluster nodes with the following verification information:

- Name of the node where verification was run
- Date and time of the last verification
- Results of the verification.

This information is stored on every available cluster node in the HACMP log file **/var/hacmp/log/clutils.log**. If the selected node became unavailable or could *not* complete cluster verification, you can detect this by the lack of a report in the **/var/hacmp/log/clutils.log** file.

In case cluster verification completes and detects some configuration errors, you are notified about the following potential problems:

- The exit status of cluster verification is communicated across the cluster along with the information about cluster verification process completion.
- Broadcast messages are sent across the cluster and displayed on **stdout**. These messages inform you about detected configuration errors.
- A **cluster_notify** event runs on the cluster and is logged in **hacmp.out** (if cluster services is running).

More detailed information is available on the node that completes cluster verification in **/var/hacmp/clverify/clverify.log**. If a failure occurs during processing, error messages and warnings clearly indicate the node and reasons for the verification failure.

Configuring Automatic Verification and Monitoring of Cluster Configuration

Make sure the **/var** filesystem on the node has enough space for the **/var/hacmp/log/clutils.log** file. For additional information, see the section The Size of the /var Filesystem May Need to be Increased in Chapter 10: Monitoring an HACMP Cluster in the *Administration Guide*.

To configure the node and specify the time where cluster verification runs automatically:

1. Enter `smit hacmp`
2. In SMIT, select **Problem Determination Tools > HACMP Verification > Automatic Cluster Configuration Monitoring**.
3. Enter field values as follows:

*** Automatic cluster configuration verification**	**Enabled** is the default.
Node name	Select one of the cluster nodes from the list. By default, the first node in alphabetical order will verify the cluster configuration. This node will be determined dynamically every time the automatic verification occurs.
***HOUR (00 - 23)**	Midnight (00) is the default. Verification runs automatically once every 24 hours at the selected hour.

4. Press Enter.
5. The changes take effect when the cluster is synchronized.

Viewing Current State

Select this option from the **Problem Determination Tools** menu to display the state of the nodes, communication interfaces, resource groups, and the local event summary for the last five events.

HACMP Log Viewing and Management

Select this option from the **Problem Determination Tools** menu to view a list of utilities related to the log files. From here you can:

- View, save or delete Event summaries
- View detailed HACMP log files
- Change or show HACMP log file parameters
- Change or show Cluster Manager log file parameters
- Change or show a cluster log file directory
- Change all Cluster Logs directory
- Collect cluster log files for problem reporting.

See Chapter 2: Using Cluster Log Files and Chapter 8: Testing an HACMP Cluster in the *Administration Guide* for complete information.

Recovering from HACMP Script Failure

Select this option from the **Problem Determination Tools** menu to recover from an HACMP script failure. For example, if a script failed because it was unable to set the hostname, the Cluster Manager reports the event failure. Once you correct the problem by setting the hostname from the command line, you must get the Cluster Manager to resume cluster processing.

The **Recover From HACMP Script Failure** menu option invokes the **/usr/es/sbin/cluster/utilities/clruncmd** command, which sends a signal to the Cluster Manager daemon **(clstrmgrES)** on the specified node, causing it to stabilize. You must again run the script manually to continue processing.

Make sure that you fix the problem that caused the script failure. You need to manually complete the remaining steps that followed the failure in the event script (see **/var/hacmp/log/hacmp.out**). Then, to resume clustering, complete the following steps to bring the HACMP event script state to EVENT COMPLETED:

1. Enter `smit hacmp`
2. In SMIT, select **Problem Determination Tools > Recover From HACMP Script Failure**.
3. Select the IP label/address for the node on which you want to run the **clruncmd** command and press Enter. The system prompts you to confirm the recovery attempt. The IP label is listed in the **/etc/hosts** file and is the name assigned to the service IP address of the node on which the failure occurred.
4. Press Enter to continue. Another SMIT panel appears to confirm the success of the script recovery.

Restoring HACMP Configuration Database from an Active Configuration

If cluster services are up and you make changes to the configuration, those changes have modified the default configuration directory (DCD). You may realize that the impact of those changes was *not* well considered and you want to undo them. Because nothing was modified in the active configuration directory (ACD), all that is needed to undo the modifications to the DCD is to restore the DCD from the ACD.

Select this option from the **Problem Determination Tools** menu to automatically save any of your changes in the Configuration Database as a snapshot with the path **/usr/es/sbin/cluster/snapshots/UserModifiedDB** before restoring the Configuration Database with the values actively being used by the Cluster Manager.

1. Enter `smit hacmp`
2. In SMIT, select **Problem Determination Tools > Restore HACMP Configuration Database from Active Configuration.**

   SMIT displays: `Are you Sure?`
3. Press Enter.

   The snapshot is saved. For complete information on snapshots, see Chapter 18: Saving and Restoring Cluster Configurations in the *Administration Guide*.

Release Locks Set by Dynamic Reconfiguration

For information on the release locks set by Dynamic Reconfiguration, see the Dynamic Reconfiguration Issues and Synchronization section in Chapter 13: Managing the Cluster Topology in the *Administration Guide*.

Clear SSA Disk Fence Registers

Select this option from the menu only in an emergency, usually only when recommended by IBM support. If SSA Disk Fencing is enabled, and a situation has occurred in which the physical disks are inaccessible by a node or a group of nodes that need access to a disk, clearing the fence registers will allow access. Once this is done, the SSA Disk Fencing algorithm will be disabled unless HACMP is restarted from all nodes.

To clear SSA Disk Fence Registers take the following steps:

1. Enter `smit hacmp`
2. In SMIT, stop cluster services (unless you are sure no contention for the disk will occur), by selecting **System Management (C-SPOC) > Manage HACMP Services > Stop Cluster Services**. For more information see the chapter on Starting and Stopping Cluster Services in the *Administration Guide*.
3. Select **Problem Determination Tools > Clear SSA Disk Fence Registers**.
4. Select the affected physical volume(s) and press Enter.
5. Restart cluster services to enable SSA disk fencing again.

HACMP Cluster Test Tool

HACMP includes the Cluster Test Tool to help you test the recovery procedures for a new cluster before it becomes part of your production environment. You can also use it to test configuration changes to an existing cluster, when the cluster is *not* in service. See the chapter on Testing an HACMP Cluster in the *Administration Guide*.

HACMP Trace Facility

Select this option from the **Problem Determination Tools** menu if the log files have no relevant information and the component-by-component investigation does *not* yield concrete results. Use the HACMP trace facility to attempt to diagnose the problem. The trace facility provides a detailed look at selected system events. Note that both the HACMP and AIX software must be running in order to use HACMP tracing.

For more information on using the trace facility, see Appendix C: HACMP Tracing. Interpreting the output generated by the trace facility requires extensive knowledge of both the HACMP software and the AIX operating system.

HACMP Event Emulation

Select this option from the **Problem Determination Tools** menu to emulate cluster events. Running this utility lets you emulate cluster events by running event scripts that produce output but do *not* affect the cluster configuration status. This allows you to predict a cluster's reaction to an event as though the event actually occurred.

The Event Emulator follows the same procedure used by the Cluster Manager given a particular event, but does *not* execute any commands that would change the status of the Cluster Manager. For descriptions of cluster events and how the Cluster Manager processes these events, see the *Planning Guide*. For more information on the cluster log redirection functionality see the chapter on Managing Resource Groups in a Cluster in the *Administration Guide*.

The event emulator runs the events scripts on every active node of a stable cluster. Output from each node is stored in an output file on the node from which you invoked the emulation. You can specify the name and location of the output file using the environment variable EMUL_OUTPUT or you can use the default output file **/var/hacmp/log/emuhacmp.out**.

Event Emulator Considerations

Keep the following cautions in mind when using the Event Emulator:

- Run only one instance of the event emulator at a time. If you attempt to start a new emulation in a cluster while an emulation is already running, the integrity of the results cannot be guaranteed. Each emulation is a stand-alone process; one emulation cannot be based on the results of a previous emulation.
- **clinfoES** must be running on all nodes.
- Add a cluster snapshot before running an emulation, just in case uncontrolled cluster events happen during emulation. Instructions for adding cluster snapshots are in the chapter on Saving and Restoring Cluster Configurations in the *Administration Guide*.
- The Event Emulator can run only event scripts that comply with the currently active configuration. For example:

- The Emulator expects to see the same environmental arguments used by the Cluster Manager; if you define arbitrary arguments, the event scripts will run, but error reports will result.
- In the case of **swap_adapter**, you must enter the **ip_label** supplied for service and non-service interfaces in the correct order, as specified in the usage statement. Both interfaces must be located on the same node at emulation time. Both must be configured as part of the same HACMP logical network.

For other events, the same types of restrictions apply. If errors occur during emulation, recheck your configuration to ensure that the cluster state supports the event to be emulated.

- The Event Emulator runs customized scripts (pre- and post-event scripts) associated with an event, but does *not* run commands within these scripts. Therefore, if these customized scripts change the cluster configuration when actually run, the outcome may differ from the outcome of an emulation.
- When emulating an event that contains a customized script, the Event Emulator uses the **ksh** flags **-n** and **-v.** The **-n** flag reads commands and checks them for syntax errors, but does *not* execute them. The **-v** flag indicates verbose mode. When writing customized scripts that may be accessed during an emulation, be aware that the other **ksh** flags may *not* be compatible with the **-n** flag and may cause unpredictable results during the emulation. See the **ksh** man page for flag descriptions.

Running Event Emulations

You can run the event emulator through SMIT.

To emulate a cluster event complete the following steps:

1. Enter `smit hacmp`.
2. In SMIT, select **Problem Determination Tools** > **HACMP Event Emulation**.

   SMIT displays a panel with options. Each option provides a different cluster event to emulate. The following sections provide more information about each option.

Emulating a Node Up Event

To emulate a Node Up event:

1. Select **Node Up Event** from the **HACMP Event Emulation** panel. SMIT displays the panel.
2. Enter the name of the node to use in the emulation.
3. Press Enter to start the emulation.

Emulating a Node Down Event

To emulate a Node Down event:

1. Select **Node Down Event** from the **HACMP Event Emulation** panel. SMIT displays the panel.
2. Enter field data as follows:

Node Name	Enter the node to use in the emulation.
Node Down Mode	Indicate the type of shutdown to emulate: • **Bring Resource Groups Offline**. The node that is shutting down releases its resources. The other nodes do *not* take over the resources of the stopped node. • **Move Resource Groups**. The node that is shutting down releases its resources. The other nodes do take over the resources of the stopped node. • **Unmanage Resource Groups**. HACMP shuts down immediately. The node that is shutting down retains control of all its resources. Applications that do *not* require HACMP daemons continue to run. Typically, you use the UNMANAGE option so that stopping the Cluster Manager does *not* interrupt users and clients. Note that enhanced concurrent volume groups do *not* accept the UNMANAGE option if they are online.

3. Press Enter to start the emulation.

Emulating a Network Up Event

To emulate a Network Up event:

1. From the **HACMP Event Emulation** panel, select **Network Up Event**. SMIT displays the panel.
2. Enter field data as follows:

Network Name	Enter the network to use in the emulation.
Node Name	(*Optional*) Enter the node to use in the emulation.

3. Press Enter to start the emulation.

Emulating a Network Down Event

To emulate a Network Down event:

1. From the **HACMP Event Emulation** panel, select **Network Down Event**. SMIT displays the panel.
2. Enter field data as follows:

Network Name	Enter the network to use in the emulation.
Node Name	(*Optional*) Enter the node to use in the emulation.

3. Press Enter to start the emulation.

Emulating a Fail Standby Event

To emulate a Fail Standby event:

1. Select **Fail Standby Event** from the **HACMP Event Emulation** panel. SMIT displays the **Fail Standby Event** panel.
2. Enter field data as follows:

Node Name	Enter the node to use in the emulation.
IP Label	Enter the IP label to use in the emulation.

3. Press Enter to start the emulation.

   The following messages are displayed on all active cluster nodes when emulating the Fail Standby and Join Standby events:

```
Adapter $ADDR is no longer available for use as a standby, due to
either a standby adapter failure or IP address takeover.

Standby adapter $ADDR is now available.
```

Emulating a Join Standby Event

To emulate a Join Standby event:

1. From the **HACMP Event Emulation** panel, select **Join Standby Event**. SMIT displays the **Join Standby Event** panel.
2. Enter field data as follows:

Node Name	Enter the node to use in the emulation.
IP Label	Enter the IP label to use in the emulation.

3. Press Enter to start the emulation.

Emulating a Swap Adapter Event

To emulate a Swap Adapter event:

1. From the **HACMP Event Emulation** panel, select **Swap Adapter Event**. SMIT displays the Swap Adapter Event panel.
2. Enter field data as follows:

Node Name	Enter the node to use in the emulation.
Network Name	Enter the network to use in the emulation.
Boot Time IP Label of available Network Interface	The name of the IP label to swap. The Boot-time IP Label must be available on the node on which the emulation is taking place.
Service Label to Move	The name of the Service IP label to swap, it must be on the and available to the same node as the Boot-time IP Label

3. Press Enter to start the emulation.

Emulating Dynamic Reconfiguration Events

To run an emulation of a Dynamic Reconfiguration event, modify the cluster configuration to reflect the configuration to be emulated and use the SMIT panels explained in this section.

Note: The Event Emulator will *not* change the configuration of a cluster device. Therefore, if your configuration contains a process that makes changes to the Cluster Manager (disk fencing, for example), the Event Emulator will *not* show these changes. This could lead to a different output, especially if the hardware devices cause a fallover.

You should add a cluster snapshot before running an emulation, just in case uncontrolled cluster events happen during emulation. Instructions for adding cluster snapshots are in the chapter on Saving and Restoring Cluster Configurations in the *Administration Guide*.

To emulate synchronizing a cluster resource event:

1. Enter `smit hacmp`
2. In SMIT select **Extended Configuration > Extended Verification and Synchronization**.
3. Enter field data as follows:

Emulate or Actual	If you set this field to **Emulate**, the synchronization will be an emulation and will *not* affect the Cluster Manager. If you set this field to **Actual**, the synchronization will actually occur, and any subsequent changes will be made to the Cluster Manager. **Emulate** is the default value. Note that these files appear only when the cluster is active.

4. Press Enter to start the emulation.

After you run the emulation, if you do *not* wish to run an actual dynamic reconfiguration, you can restore the original configuration using the SMIT panel option **Problem Determination Tools > Restore System Default Configuration from Active Configuration**.

HACMP Error Notification

For complete information on setting up both AIX and HACMP error notification, see the chapter on Tailoring AIX for HACMP, in the *Installation Guide*.

Opening a SMIT Session on a Node

As a convenience while troubleshooting your cluster, you can open a SMIT session on a remote node from within the **Problem Determination Tool** SMIT panel.

To open a SMIT session on a remote node:

1. Select the **Problem Determination Tools** > **Open a SMIT Session on a Node** option. SMIT displays a list of available cluster nodes.
2. Select the node on which you wish to open the SMIT session and press Enter.

Configuring Cluster Performance Tuning

Cluster nodes sometimes experience extreme performance problems such as large I/O transfers, excessive error logging, or lack of memory. When this happens, the Cluster Manager can be starved for CPU time and it may *not* reset the deadman switch within the time allotted. Misbehaved applications running at a priority higher than the Cluster Manager can also cause this problem.

The deadman switch is the AIX kernel extension that halts a node when it enters a hung state that extends beyond a certain time limit. This enables another node in the cluster to acquire the hung node's resources in an orderly fashion, avoiding possible contention problems. If the deadman switch is *not* reset in time, it can cause a system panic and dump under certain cluster conditions.

Setting the following tuning parameters correctly may avoid some of the performance problems noted above. To prevent the possibility of having to change the HACMP Network Modules Failure Detection Rate, it is highly recommended to first set the following two AIX parameters:

- AIX high and low watermarks for I/O pacing
- AIX **syncd** frequency rate.

Set the two AIX parameters on each cluster node.

You may also set the following HACMP network tuning parameters for each type of network:

- Failure Detection Rate
- Grace Period.

You can configure these related parameters directly from HACMP SMIT.

Network module settings are propagated to all nodes when you set them on one node and then synchronize the cluster topology.

Setting I/O Pacing

In some cases, you can use I/O pacing to tune the system so that system resources are distributed more equitably during large disk writes. However, the results of tuning I/O pacing are highly dependent on each system's specific configuration and I/O access characteristics.

I/O pacing can help ensure that the HACMP Cluster Manager continues to run even during large disk writes. In some situations, it can help prevent DMS timeouts.

Note: Setting I/O pacing can significantly reduce system performance and throughput.

Remember, I/O pacing and other tuning parameters should only be set after a system performance analysis indicates that doing so will lead to both the desired and acceptable side effects.

If you experience workloads that generate large disk writes or intense amounts of disk traffic, contact IBM for recommendations on choices of tuning parameters that will both allow HACMP to function, and provide acceptable performance. To contact IBM, open a Program Management Report (PMR) requesting performance assistance, or follow other established procedures for contacting IBM.

Although the most efficient high- and low-water marks vary from system to system, an initial high-water mark of **33** and a low-water mark of **24** provides a good starting point. These settings only slightly reduce write times and consistently generate correct fallover behavior from the HACMP software.

See the *AIX Performance Monitoring & Tuning Guide* for more information on I/O pacing.

To change the I/O pacing settings, do the following on each node:

1. Enter `smit hacmp`
2. In SMIT, select **Extended Configuration > Extended Performance Tuning Parameters Configuration > Change/Show I/O Pacing** and press Enter.
3. Configure the field values with the recommended HIGH and LOW watermarks:

HIGH water mark for pending write I/Os per file	**33** is recommended for most clusters. Possible values are 0 to 32767.
LOW watermark for pending write I/Os per file	**24** is recommended for most clusters. Possible values are 0 to 32766.

Setting Syncd Frequency

The **syncd** setting determines the frequency with which the I/O disk-write buffers are flushed. Frequent flushing of these buffers reduces the chance of deadman switch time-outs.

The AIX default value for **syncd** as set in **/sbin/rc.boot** is 60. It is recommended to change this value to 10. Note that the I/O pacing parameters setting should be changed first.

To change the **syncd** frequency setting, do the following on each node:

1. Enter `smit hacmp`
1. In SMIT, select **Extended Configuration > Extended Performance Tuning Parameters > Change/Show syncd frequency** and press Enter.
2. Configure the field values with the recommended **syncd** frequency:

syncd frequency (in seconds)	**10** is recommended for most clusters. Possible values are 0 to 32767.

Changing the Failure Detection Rate of a Network Module after the Initial Configuration

If you want to change the failure detection rate of a network module, either change the tuning parameters of a network module to predefined values of **Fast**, **Normal** and **Slow**, or set these attributes to custom values.

Also, use the custom tuning parameters to change the baud rate for TTYs if you are using RS232 networks that might *not* handle the default baud rate of 38400.

For more information, see the Changing the Configuration of a Network Module section in the chapter on Managing the Cluster Topology in the *Administration Guide*.

Resetting HACMP Tunable Values

In HACMP 5.2 and up, you can change the settings for a list of tunable values that were altered during cluster maintenance and reset them to their default settings, or installation-time cluster settings. The *installation-time* cluster settings are equal to the values that appear in the cluster after installing HACMP from scratch.

Resetting cluster tunables is useful when an administrative change has not produced the desired results, and you want to return to the default values. While this may not produce the optimum configuration, it is very likely to produce a working one and allows you to make further changes from a known base.

Note: Resetting the tunable values *does not* change any other aspects of the configuration, while installing HACMP removes all user-configured configuration information including nodes, networks, and resources.

Prerequisites and Limitations

You can change and reset HACMP tunable values to their default values under the following conditions:

- Before resetting HACMP tunable values, HACMP takes a cluster snapshot. After the values have been reset to defaults, if you want to go back to your customized cluster settings, you can restore them with the cluster snapshot. HACMP saves snapshots of the last ten configurations in the default cluster snapshot directory, **/usr/es/sbin/cluster/snapshots**, with the name **active.x.odm**, where x is a digit between 0 and 9, with 0 being the most recent.
- Stop cluster services on all nodes before resetting tunable values. HACMP prevents you from resetting tunable values in a running cluster.

In some cases, HACMP cannot differentiate between user-configured information and discovered information, and *does not* reset such values. For example, you may enter a service label and HACMP automatically discovers the IP address that corresponds to that label. In this case, HACMP does not reset the service label or the IP address. The cluster verification utility detects if these values do *not* match.

The **clsnapshot.log** file in the snapshot directory contains log messages for this utility. If any of the following scenarios are run, then HACMP cannot revert to the previous configuration:

- **cl_convert** is run automatically
- **cl_convert** is run manually
- **clconvert_snapshot** is run manually. The **clconvert_snapshot** utility is *not* run automatically, and must be run from the command line to upgrade cluster snapshots when migrating from HACMP (HAS) to HACMP 5.1 or greater.

Listing Tunable Values

You can change and reset the following list of tunable values:

- User-supplied information.
  - Network module tuning parameters such as failure detection rate, grace period and heartbeat rate. HACMP resets these parameters to their installation-time default values.

- Cluster event customizations such as all changes to cluster events. Note that resetting changes to cluster events does *not* remove any files or scripts that the customization used, it only removes the knowledge HACMP has of pre- and post-event scripts.
- Cluster event rule changes made to the event rules database are reset to the installation-time default values.
- HACMP command customizations made to the default set of HACMP commands are reset to the installation-time defaults.

- Automatically generated and discovered information, generally users cannot see this information. HACMP rediscovers or regenerates this information when the cluster services are restarted or during the next cluster synchronization.

  HACMP resets the following:
  - Local node names stored in the cluster definition database
  - Netmasks for all cluster networks
  - Netmasks, interface names and aliases for disk heartbeating (if configured) for all cluster interfaces
  - SP switch information generated during the latest **node_up** event (this information is regenerated at the next **node_up** event)
  - Instance numbers and default log sizes for the RSCT subsystem.

Resetting HACMP Tunable Values Using SMIT

To reset cluster tunable values to default values:

1. Enter `smit hacmp`.
2. In SMIT, select **Extended Configuration > Extended Topology Configuration > Configure an HACMP Cluster > Reset Cluster Tunables** and press Enter.

Use this option to reset all the tunables (customizations) made to the cluster. For a list of the tunable values that will change, see the section Listing Tunable Values. Using this option returns all tunable values to their default values but does *not* change the cluster configuration. HACMP takes a snapshot file before resetting. You can choose to have HACMP synchronize the cluster when this operation is complete.

3. Select the options as follows and press Enter:

Synchronize Cluster Configuration	If you set this option to **yes**, HACMP synchronizes the cluster after resetting the cluster tunables.

4. HACMP asks: `"Are you sure?"`
5. Press Enter.

   HACMP resets all the tunable values to their original settings and removes those that should be removed (such as the nodes' knowledge about customized pre- and post-event scripts).

Resetting HACMP Tunable Values using the Command Line

We recommend that you use the SMIT interface to reset the cluster tunable values. The **clsnapshot -t** command also resets the cluster tunables. This command is intended for use by IBM support. See the man page for more information.

Sample Custom Scripts

Two situations where it is useful to run custom scripts are illustrated here:

- Making **cron** jobs highly available
- Making print queues highly available.

Making cron jobs Highly Available

To help maintain the HACMP environment, you need to have certain **cron** jobs execute only on the cluster node that currently holds the resources. If a **cron** job executes in conjunction with a resource or application, it is useful to have that **cron** entry fallover along with the resource. It may also be necessary to remove that **cron** entry from the **cron** table if the node no longer possesses the related resource or application.

The following example shows one way to use a customized script to do this:

The example cluster is a two node hot standby cluster where node1 is the primary node and node2 is the backup. Node1 normally owns the shared resource group and application. The application requires that a **cron** job be executed once per day but only on the node that currently owns the resources.

To ensure that the job will run even if the shared resource group and application fall over to node2, create two files as follows:

1. Assuming that the root user is executing the **cron** job, create the file **root.resource** and another file called **root.noresource** in a directory on a non-shared file system on node1. Make these files resemble the **cron** tables that reside in the directory **/var/spool/crontabs**.

   The **root.resource** table should contain all normally executed system entries, and all entries pertaining to the shared resource or application.

   The **root.noresource** table should contain all normally executed system entries but should *not* contain entries pertaining to the shared resource or application.

2. Copy the files to the other node so that both nodes have a copy of the two files.
3. On both systems, run the following command at system startup:

   ```
   crontab root.noresource
   ```

   This will ensure that the **cron** table for root has only the “no resource” entries at system startup.

4. You can use either of two methods to activate the *root.resource* **cron** table. The first method is the simpler of the two.

- Run **crontab root.resource** as the last line of the application start script. In the application stop script, the first line should then be **crontab root.noresource**. By executing these commands in the application start and stop scripts, you are ensured that they will activate and deactivate on the proper node at the proper time.
- Run the **crontab** commands as a post_event to node_up_complete and node_down_complete.
  - Upon node_up_complete on the primary node, run **crontab root.resources**.
  - On node_down_complete run **crontab root.noresources**.

  The takeover node must also use the event handlers to execute the correct **cron** table. Logic must be written into the node_down_complete event to determine if a takeover has occurred and to run the **crontab root.resources** command. On a reintegration, a pre-event to node_up must determine if the primary node is coming back into the cluster and then run a **crontab root.noresource** command.

When HACMP Does Not Save the AIX Environment for node_up_complete events

If your scripts to start or stop application servers depend on any information in **/etc/environment**, you should explicitly define that information in the scripts.

PATH and NLSPATH are two commonly needed variables that are not set to the values contained in **/etc/environment** during the execution of application start and stop scripts.

For example, add this line to the application scripts:

```
export PATH=/usr/bin:/bin:/sbin:/usr/sbin:/usr/local/bin
```

Or add this Korn Shell example to the **/etc/environment** script:

```
. /etc/environment
```

Making Print Queues Highly Available

In the event of a fallover, the currently queued print jobs can be saved and moved over to the surviving node.

The print spooling system consists of two directories: /**var/spool/qdaemon** and **/var/spool/lpd/qdir**. One directory contains files containing the data (content) of each job. The other contains the files consisting of information pertaining to the print job itself. When jobs are queued, there are files in each of the two directories. In the event of a fallover, these directories do *not* normally fallover and therefore the print jobs are lost.

The solution for this problem is to define two file systems on a shared volume group. You might call these file systems **/prtjobs** and **/prtdata**. When HACMP starts, these file systems are mounted over **/var/spool/lpd/qdir** and **/var/spool/qdaemon**.

Write a script to perform this operation as a post event to node_up. The script should do the following:

- Stop the print queues
- Stop the print queue daemon
- Mount **/prtjobs** over **/var/spool/lpd/qdir**
- Mount **/prtdata** over **/var/spool/qdaemon**

- Restart the print queue daemon
- Restart the print queues.

In the event of a fallover, the surviving node will need to do the following:

- Stop the print queues
- Stop the print queue daemon
- Move the contents of **/prtjobs** into **/var/spool/lpd/qdir**
- Move the contents of **/prtdata** into /**var/spool/qdaemon**
- Restart the print queue daemon
- Restart the print queues.

To do this, write a script called as a post-event to **node_down_complete** on the takeover. The script needs to determine if the **node_down** is from the primary node.

Where You Go from Here

Chapter 2: Using Cluster Log Files describes how to use the HACMP cluster log files to troubleshoot the cluster.

For more information on using HACMP and AIX utilities see Chapter 3: Investigating System Components and Solving Common Problems.

Chapter 2: Using Cluster Log Files

This chapter explains how to use the HACMP cluster log files to troubleshoot the cluster. It also includes sections on managing parameters for some of the logs.

Major sections of the chapter include:

- Viewing HACMP Cluster Log Files
- Tracking Resource Group Parallel and Serial Processing in the hacmp.out File
- Managing a Node's HACMP Log File Parameters
- Logging for clcomd
- Redirecting HACMP Cluster Log Files.

Viewing HACMP Cluster Log Files

Your first approach to diagnosing a problem affecting your cluster should be to examine the cluster log files for messages output by the HACMP subsystems. These messages provide valuable information for understanding the current state of the cluster. The following sections describe the types of messages output by the HACMP software and the log files into which the system writes these messages.

For most troubleshooting, the **/var/hacmp/log/hacmp.out** file will be the most helpful log file. Resource group handling has been enhanced in recent releases and the **hacmp.out** file has been expanded to capture more information on the activity and location of resource groups after cluster events. For instance, the **hacmp.out** file captures details of resource group parallel processing that other logs (such as the cluster history log) cannot report. The event summaries included in this log make it easier to see quickly what events have occurred recently in the cluster.

Reviewing Cluster Message Log Files

The HACMP software writes the messages it generates to the system console and to several log files. Each log file contains a different subset of messages generated by the HACMP software. When viewed as a group, the log files provide a detailed view of all cluster activity.

The following list describes the log files into which the HACMP software writes messages and the types of cluster messages they contain. The list also provides recommendations for using the different log files. Note that the default log directories are listed here; you have the option of redirecting some log files to a chosen directory. For more information about how to redirect cluster log files see the section . If you have redirected any logs, check the appropriate location.

system error log	Contains time-stamped, formatted messages from all AIX subsystems, including scripts and daemons. For information about viewing this log file and interpreting the messages it contains, see the section Understanding the System Error Log. **Recommended Use:** Because the **system error log** contains time-stamped messages from many other system components, it is a good place to correlate cluster events with system events.
tmp/clconvert.log	Contains a record of the conversion progress when upgrading to a recent HACMP release. The installation process runs the **cl_convert** utility and creates the **/tmp/clconvert.log** file. **Recommended Use**: View the **clconvert.log** to gauge conversion success when running **cl_convert** from the command line. For detailed information on the **cl_convert** utility see the chapter on Upgrading an HACMP Cluster, in the *Installation Guide*.
/usr/es/sbin/cluster/snapshots/ clsnapshot.log	Contains logging information from the snapshot utility of HACMP, and information about errors found and/or actions taken by HACMP for resetting cluster tunable values.
/usr/es/sbin/cluster/wsm/logs/ wsm_smit.log	All operations of the WebSMIT interface are logged to the **wsm_smit.log** file and are equivalent to the logging done with **smitty -v**. Script commands are also captured in the **wsm_smit.script** log file. **wsm_smit** log files are created by the CGI scripts using a relative path of **<../logs>**. If you copy the CGI scripts to the default location for the IBM HTTP Server, the final path to the logs is */usr/IBMIHS/logs*. The location of the WebSMIT log files cannot be modified. Like log files **smit.log** and **smit.script**, new logging entries are appended to the end of the file, and you need to control their size and backup. There is no default logging of the cluster status display, although logging can be enabled through the **wsm_clstat.com** configuration file.
/var/ha/log/grpglsm	Contains time-stamped messages in ASCII format. These track the execution of internal activities of the RSCT Group Services Globalized Switch Membership daemon. IBM support personnel use this information for troubleshooting. The file gets trimmed regularly. Therefore, please save it promptly if there is a chance you may need it.

/var/ha/log/grpsvcs	Contains time-stamped messages in ASCII format. These track the execution of internal activities of the RSCT Group Services daemon. IBM support personnel use this information for troubleshooting. The file gets trimmed regularly. Therefore, please save it promptly if there is a chance you may need it.
/var/ha/log/topsvcs.<*filename*>	Contains time-stamped messages in ASCII format. These track the execution of internal activities of the RSCT Topology Services daemon. IBM support personnel use this information for troubleshooting. The file gets trimmed regularly. Therefore, please save it promptly if there is a chance you may need it.
/var/hacmp/adm/cluster.log	Contains time-stamped, formatted messages generated by HACMP scripts and daemons. For information about viewing this log file and interpreting its messages, see the following section Understanding the cluster.log File. **Recommended Use:** Because this log file provides a high-level view of current cluster status, check this file first when diagnosing a cluster problem.
/var/hacmp/adm/history/ cluster.*mmddyyyy*	Contains time-stamped, formatted messages generated by HACMP scripts. The system creates a cluster history file every day, identifying each file by its file name extension, where *mm* indicates the month, *dd* indicates the day, and *yyyy* the year. For information about viewing this log file and interpreting its messages, see the section Understanding the Cluster History Log File. **Recommended Use:** Use the cluster history log files to get an extended view of cluster behavior over time. Note that this log is *not* a good tool for tracking resource groups processed in parallel. In parallel processing, certain steps formerly run as separate events are now processed differently and these steps will *not* be evident in the cluster history log. Use the **hacmp.out** file to track parallel processing activity.
/var/hacmp/clcomd/ clcomddiag.log	Contains time-stamped, formatted, diagnostic messages generated by **clcomd**. **Recommended Use:** Information in this file is for IBM Support personnel.
/var/hacmp/log/autoverify.log	Contains logging for Automatic Cluster Verification.

/var/hacmp/log/clavan.log	Contains the state transitions of applications managed by HACMP. For example, when each application managed by HACMP is started or stopped and when the node stops on which an application is running. Each node has its own instance of the file. Each record in the **clavan.log** file consists of a single line. Each line contains a fixed portion and a variable portion: **Recommended Use:** By collecting the records in the **clavan.log** file from every node in the cluster, a utility program can determine how long each application has been up, as well as compute other statistics describing application availability time.
/var/hacmp/log/clinfo.log **/var/hacmp/log/clinfo.log.n, n=1,..,7**	The **clinfo.log** file records the output generated by the event scripts as they run. This information supplements and expands upon the information in the **/var/hacmp/log/hacmp.out** file. You can install Client Information (Clinfo) services on both client and server systems – client systems (cluster.es.client) will *not* have any HACMP ODMs (for example HACMPlogs) or utilities (for example clcycle); therefore, the Clinfo logging will *not* take advantage of cycling or redirection. The default debug level is 0 or “off”. You can enable logging using command line flags. Use the clinfo -l flag to change the log file name.
/var/hacmp/log/clstrmgr.debug **/var/hacmp/log/clstrmgr.debug.n, n=1,..,7**	Contains time-stamped, formatted messages generated by the **clstrmgrES** daemon. The default messages are verbose and are typically adequate for troubleshooting most problems, however IBM support may direct you to enable additional debugging. **Recommended Use:** Information in this file is for IBM Support personnel.
/var/hacmp/log/ clstrmgr.debug.long **/var/hacmp/log/ clstrmgr.debug.long.n, n=1,..,7**	Contains high-level logging of cluster manager activity, in particular its interaction with other components of HACMP and with RSCT, which event is currently being run, and information about resource groups (for example, their state and actions to be performed, such as acquiring or releasing them during an event.
/var/hacmp/log/cspoc.log	Contains time-stamped, formatted messages generated by HACMP C-SPOC commands. The **cspoc.log** file resides on the node that invokes the C-SPOC command. **Recommended Use:** Use the C-SPOC log file when tracing a C-SPOC command’s execution on cluster nodes.

/var/hacmp/log/cspoc.log.long	Contains a high-level of logging for the C-SPOC utility – commands and utilities that have been invoked by C-SPOC on specified nodes and their return status.
/var/hacmp/log/ cspoc.log.remote	Contains logging of the execution of C-SPOC commands on remote nodes with **ksh** option **xtrace** enabled (**set -x**).
/var/hacmp/log/emuhacmp.out	Contains time-stamped, formatted messages generated by the HACMP Event Emulator. The messages are collected from output files on each node of the cluster, and cataloged together into the **emuhacmp.out** log file. In verbose mode (recommended), this log file contains a line-by-line record of every event emulated. Customized scripts within the event are displayed, but commands within those scripts are *not* executed.
/var/hacmp/log/hacmp.out /var/hacmp/log/hacmp.out.n n=1,..,7	Contains time-stamped, formatted messages generated by HACMP scripts on the current day. In verbose mode (recommended), this log file contains a line-by-line record of every command executed by scripts, including the values of all arguments to each command. An event summary of each high-level event is included at the end of each event's details. For information about viewing this log and interpreting its messages, see the section Understanding the hacmp.out Log File. **Recommended Use:** Because the information in this log file supplements and expands upon the information in the **/var/hacmp/adm/cluster.log** file, it is the primary source of information when investigating a problem. **Note:** With recent changes in the way resource groups are handled and prioritized in fallover circumstances, the **hacmp.out** file and its event summaries have become even more important in tracking the activity and resulting location of your resource groups. In HACMP releases prior to 5.2, non-recoverable event script failures result in the **event_error** event being run on the cluster node where the failure occurred. The remaining cluster nodes do *not* indicate the failure. With HACMP 5.2 and up, all cluster nodes run the **event_error** event if any node has a fatal error. All nodes log the error and call out the failing node name in the **hacmp.out** log file.
/var/hacmp/log/oraclesa.log	Contains logging of the Smart Assist for Oracle facility.
/var/hacmp/log/sa.log	Contains logging of the Smart Assist facility.

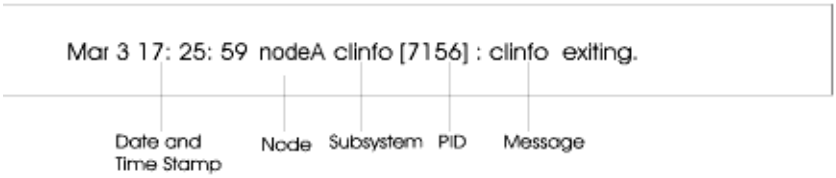
/var/hacmp/clcomd/clcomd.log	Contains time-stamped, formatted messages generated by Cluster Communications daemon (**clcomd)** activity. The log shows information about incoming and outgoing connections, both successful and unsuccessful. Also displays a warning if the file permissions for **/usr/es/sbin/cluster/etc/rhosts** are *not* set correctly—users on the system should *not* be able to write to the file. **Recommended Use:** Use information in this file to troubleshoot inter-node communications, and to obtain information about attempted connections to the daemon (and therefore to HACMP).
/var/hacmp/log/ clconfigassist.log	Contains debugging information for the Two-Node Cluster Configuration Assistant. The Assistant stores up to ten copies of the numbered log files to assist with troubleshooting activities.
/var/hacmp/clverify/clverify.log	The **clverify.log** file contains the verbose messages output by the cluster verification utility. The messages indicate the node(s), devices, command, etc. in which any verification error occurred. For complete information see Chapter 7: Verifying and Synchronizing a Cluster Configuration in the *Administration Guide*.
/var/hacmp/log/clutils.log	Contains information about the date, time, results, and which node performed an automatic cluster configuration verification. It also contains information for the file collection utility, the two-node cluster configuration assistant, the cluster test tool and the OLPW conversion tool.
/var/hacmp/log/ cl_testtool.log	Includes excerpts from the **hacmp.out** file. The Cluster Test Tool saves up to three log files and numbers them so that you can compare the results of different cluster tests. The tool also rotates the files with the oldest file being overwritten
/var/hacmp/log/migration.log	Contains a high level of logging of cluster activity while the cluster manager on the local node operates in a migration state. All actions pertaining to the cluster manager follow the internal migration protocol.

Understanding the cluster.log File

The **/var/hacmp/adm/cluster.log** file is a standard text file. When checking this file, first find the most recent error message associated with your problem. Then read back through the log file to the first message relating to that problem. Many error messages cascade from an initial error that usually indicates the problem source.

Format of Messages in the cluster.log File

The entries in the **/var/hacmp/adm/cluster.log f**ile use the following format:



Each entry has the following information:

Date and Time stamp	The day and time on which the event occurred.
Node	The node on which the event occurred.
Subsystem	The HACMP subsystem that generated the event. The subsystems are identified by the following abbreviations: • **clstrmgrES—**the Cluster Manager daemon • **clinfoES—**the Cluster Information Program daemon • **HACMP for AIX—**startup and reconfiguration scripts.
PID	The process ID of the daemon generating the message (*not* included for messages output by scripts).
Message	The message text.

The entry in the previous example indicates that the Cluster Information program (**clinfoES**) stopped running on the node named *nodeA* at 5:25 P.M. on March 3.

Because the **/var/hacmp/adm/cluster.log** file is a standard ASCII text file, you can view it using standard AIX file commands, such as the **more** or **tail** commands. However, you can also use the SMIT interface. The following sections describe each of the options.

Viewing the cluster.log File Using SMIT

To view the **/var/hacmp/adm/cluster.log** file using SMIT:

1. Enter `smit hacmp`
2. In SMIT, select **Problem Determination Tools > HACMP Log Viewing and Management** > **View Detailed HACMP Log Files** and press Enter.
3. Select **Scan the HACMP for AIX System Log** and press Enter. This option references the **/var/hacmp/adm/cluster.log** file.

Note: You can select to either *scan* the contents of the **cluster.log** file as it exists, or you can *watch* an active log file as new events are appended to it in real time. Typically, you scan the file to try to find a problem that has already occurred; you watch the file as you test a solution to a problem to determine the results.

Understanding the hacmp.out Log File

The **/var/hacmp/log/hacmp.out** file is a standard text file. The system cycles **hacmp.out** log file seven times. Each copy is identified by a number appended to the file name. The most recent log file is named **/var/hacmp/log/hacmp.out**; the oldest version of the file is named **/var/hacmp/log/hacmp.out.7**.

Given the recent changes in the way resource groups are handled and prioritized in fallover circumstances, the **hacmp.out** file and its event summaries have become even more important in tracking the activity and resulting location of your resource groups.

You can customize the wait period before a warning message appears. Since this affects how often the **config_too_long** message is posted to the log, the **config_too_long** console message may *not* be evident in every case where a problem exists. See details below in the Config_too_long Message in the hacmp.out File section.

In HACMP releases prior to 5.2, non-recoverable event script failures result in the **event_error** event being run on the cluster node where the failure occurred. The remaining cluster nodes do *not* indicate the failure. With HACMP 5.2 and up, all cluster nodes run the **event_error** event if any node has a fatal error. All nodes log the error and call out the failing node name in the **hacmp.out** log file.

When checking the **hacmp.out** file, search for EVENT FAILED messages. These messages indicate that a failure has occurred. Then, starting from the failure message, read back through the log file to determine exactly what went wrong. The **hacmp.out** log file provides the most important source of information when investigating a problem.

Note: With HACMP 5.2 and up, EVENT_FAILED_NODE is set to the name of the node where the event failed.

Event Preambles

When an event processes resource groups with dependencies or with HACMP/XD replicated resources, an event preamble is included in the **hacmp.out** file. This preamble shows you the logic the Cluster Manager will use to process the event in question. See the sample below.

```
               HACMP Event Preamble
------------------------------------------------------------------------

Node Down Completion Event has been enqueued.
------------------------------------------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                        HACMP Event Preamble
Action:         Resource:
------------------------------------------------------------------------
Enqueued rg_move acquire event for resource group rg3.

Enqueued rg_move release event for resource group rg3.

Enqueued rg_move secondary acquire event for resource group 'rg1'.

Node Up Completion Event has been enqueued.
------------------------------------------------------------------------
```

Event Summaries

Event summaries that appear at the end of each event's details make it easier to check the **hacmp.out** file for errors. The event summaries contain pointers back to the corresponding event, which allow you to easily locate the output for any event. See the section Verbose Output Example with Event Summary for an example of the output.

You can also view a compilation of only the event summary sections pulled from current and past **hacmp.out** files. The option for this display is found on the **Problem Determination Tools > HACMP Log Viewing and Management > View/Save/Remove Event Summaries > View Event Summaries** SMIT panel. For more detail, see the section Viewing Compiled hacmp.out Event Summaries later in this chapter.

hacmp.out in HTML Format

You can view the **hacmp.out** log file in HTML format by setting formatting options on the **Problem Determination Tools > HACMP Log Viewing and Management > Change/Show HACMP Log File Parameters** SMIT panel. For instructions see the section Setting the Level and Format of Information Recorded in the hacmp.out File.

Resource Group Acquisition Failures and Volume Group Failures in hacmp.out

Reported resource group acquisition failures (failures indicated by a non-zero exit code returned by a command) are tracked in **hacmp.out**. This information includes:

- The start and stop times for the event
- Which resource groups were affected (acquired or released) as a result of the event
- In the case of a failed event, an indication of which resource action failed.

You can track the path the Cluster Manager takes as it tries to keep resources available.

In addition, the automatically configured AIX Error Notification method that runs in the case of a volume group failure writes the following information in the **hacmp.out** log file:

- AIX error label and ID for which the method was launched
- The name of the affected resource group
- The node's name on which the error occurred.

Messages for Resource Group Recovery Upon node_up

The **hacmp.out** file, event summaries, and **clstat** include information and messages about resource groups in the ERROR state that attempted to get online on a joining node, or on a node that is starting up.

Similarly, you can trace the cases in which the acquisition of such a resource group has failed, and HACMP launched an **rg_move** event to move the resource group to another node in the nodelist. If, as a result of consecutive **rg_move** events through the nodes, a non-concurrent resource group still failed to get acquired, HACMP adds a message to the **hacmp.out** file.

"Standby" Events Reported for Networks Using Aliases

When you add a network interface on a network using aliases, the actual event that runs in this case is called **join_interface**. This is reflected in the **hacmp.out** file. However, such networks by definition do *not* have standby interfaces defined, so the event that is being run in this case

simply indicates that a network interface joins the cluster. Similarly, when a network interface failure occurs, the actual event that is run in is called **fail_interface**. This is also reflected in the **hacmp.out** file. Remember that the event that is being run in this case simply indicates that a network interface on the given network has failed.

Resource Group Processing Messages in the hacmp.out File

The **hacmp.out** file allows you to fully track how resource groups have been processed in HACMP. This section provides a brief description, for detailed information and examples of event summaries with *job types*, see the section Tracking Resource Group Parallel and Serial Processing in the hacmp.out File later in this chapter.

For each resource group that has been processed by HACMP, the software sends the following information to the **hacmp.out** file:

- Resource group name
- Script name
- Name of the command that is being executed.

The general pattern of the output is:

```
resource_group_name:script_name [line number] command line
```

In cases where an event script does *not* process a specific resource group, for instance, in the beginning of a **node_up** event, a resource group's name cannot be obtained. In this case, the resource group's name part of the tag is blank.

For example, the **hacmp.out** file may contain either of the following lines:

```
cas2:node_up_local[199] set_resource_status ACQUIRING
:node_up[233] cl_ssa_fence up stan
```

In addition, references to the individual resources in the event summaries in the **hacmp.out** file contain reference tags to the associated resource groups. For instance:

```
Mon.Sep.10.14:54:49.EDT 2003.cl _swap_IP_address.192.168.1.1.cas2.ref
```

Config_too_long Message in the hacmp.out File

You can customize the waiting period before a **config_too_long** message is sent.

For each cluster event that does *not* complete within the specified event duration time, **config_too_long** messages are logged in the **hacmp.out** file and sent to the console according to the following pattern:

- The first five **config_too_long** messages appear in the **hacmp.out** file at 30-second intervals
- The next set of five messages appears at an interval that is double the previous interval until the interval reaches one hour
- These messages are logged every hour until the event completes or is terminated on that node.

For more information on customizing the event duration time before receiving a **config_too_long** warning message, see the chapter on Planning for Cluster Events in the *Planning Guide*.

Non-Verbose and Verbose Output of the hacmp.out Log File

You can select either verbose or non-verbose output.

Non-Verbose Output

In non-verbose mode, the **hacmp.out** log contains the start, completion, and error notification messages output by all HACMP scripts. Each entry contains the following information:

Date and Time Stamp	The day and time on which the event occurred.
Message	Text that describes the cluster activity.
Return Status	Messages that report failures include the status returned from the script. This information is *not* included for scripts that complete successfully.
Event Description	The specific action attempted or completed on a node, file system, or volume group.

Verbose Output

In verbose mode, the **hacmp.out** file also includes the values of arguments and flag settings passed to the scripts and commands.

Verbose Output Example with Event Summary

Some events (those initiated by the Cluster Manager) are followed by event summaries, as shown in these excerpts:

```
....
Mar 25 15:20:30 EVENT COMPLETED: network_up alcuin tmssanet_alcuin_bede

                        HACMP Event Summary
Event: network_up alcuin tmssanet_alcuin_bede
Start time: Tue Mar 25 15:20:30 2003

End time: Tue Mar 25 15:20:30 2003

Action:         Resource:                       Script Name:
----------------------------------------------------------------------------
No resources changed as a result of this event
----------------------------------------------------------------------------
```

Event Summary for the Settling Time

CustomRG has a settling time configured. A lower priority node joins the cluster:

```
Mar 25 15:20:30 EVENT COMPLETED: node_up alcuin

                        HACMP Event Summary
Event: node_up alcuin
Start time: Tue Mar 25 15:20:30 2003

End time: Tue Mar 25 15:20:30 2003

Action:       Resource:                       Script Name:
------------------------------------------------------------

No action taken on resource group 'CustomRG'.
The Resource Group 'CustomRG' has been configured
to use 20 Seconds Settling Time. This group will be
processed when the timer expires.
----------------------------------------------------------------------------
```

Event Summary for the Fallback Timer

CustomRG has a daily fallback timer configured to fall back on 22 hrs 10 minutes. The resource group is on a lower priority node (bede). Therefore, the timer is ticking; the higher priority node (alcuin) joins the cluster:

```
The message on bede ...

Mar 25 15:20:30 EVENT COMPLETED: node_up alcuin

                        HACMP Event Summary
Event: node_up alcuin
Start time: Tue Mar 25 15:20:30 2003

End time: Tue Mar 25 15:20:30 2003

Action:       Resource:                       Script Name:
------------------------------------------------------------

No action taken on resource group 'CustomRG'.
The Resource Group 'CustomRG' has been configured
to fallback on Mon Mar 25 22:10:00 2003
------------------------------------------------------------------------
The message on alcuin ...
Mar 25 15:20:30 EVENT COMPLETED: node_up alcuin

                        HACMP Event Summary
Event: node_up alcuin
Start time: Tue Mar 25 15:20:30 2003

End time: Tue Mar 25 15:20:30 2003

Action:       Resource:                       Script Name:
------------------------------------------------------------

The Resource Group 'CustomRG' has been configured
to fallback using daily1 Timer Policy
------------------------------------------------------------------------
```

Viewing the hacmp.out File using SMIT

To view the **/var/hacmp/log/hacmp.out** file using SMIT:

1. Enter `smit hacmp`
2. In SMIT, select **Problem Determination Tools > HACMP Log Viewing and Management** > **View Detailed HACMP Log Files** and press Enter.
3. On the **View Detailed HACMP Log Files** menu, you can select to either *scan* the contents of the**/var/hacmp/log/hacmp.out** file or *watch* as new events are appended to the log file. Typically, you will scan the file to try to find a problem that has already occurred and then watch the file as you test a solution to the problem. In the menu, the **/var/hacmp/log/hacmp.out** file is referred to as the HACMP Script Log File.
4. Select **Scan the HACMP Script Log File** and press Enter.
5. Select a script log file and press Enter.

Setting the Level and Format of Information Recorded in the hacmp.out File

Note: These preferences take place as soon as you set them.

To set the level of information recorded in the **/var/hacmp/log/hacmp.out** file:

1. Enter `smit hacmp`
2. In SMIT, select **Problem Determination Tools > HACMP Log Viewing and Management > Change/Show HACMP Log File Parameters**.

   SMIT prompts you to specify the name of the cluster node you want to modify. Runtime parameters are configured on a per-node basis.
3. Type the node name and press Enter.

   SMIT displays the **HACMP Log File Parameters** panel.
4. To obtain verbose output, set the value of the **Debug Level** field to **high.**
5. To change the **hacmp.out** display format, select **Formatting options for hacmp.out.** Select a node and set the formatting to **HTML (Low)**, **HTML (High)**, **Default (None)**, or **Standard**.

   Note: If you set your formatting options for **hacmp.out** to **Default (None)**, then no event summaries will be generated. For information about event summaries, see the section Viewing Compiled hacmp.out Event Summaries.
6. To change the level of debug information, set the value of **New Cluster Manager debug level** field to either **Low** or **High**.

Viewing Compiled hacmp.out Event Summaries

In the **hacmp.out** file, event summaries appear after those events that are initiated by the Cluster Manager. For example, **node_up** and **node_up_complete** and related subevents such as **node_up_local** and **node_up_remote_complete**. Note that event summaries do *not* appear for all events; for example, when you move a resource group through SMIT.

The **View Event Summaries** option displays a compilation of all event summaries written to a node's **hacmp.out** file. This utility can gather and display this information even if you have redirected the **hacmp.out** file to a new location. You can also save the event summaries to a file of your choice instead of viewing them via SMIT.

Note: Event summaries pulled from the **hacmp.out** file are stored in the **/usr/es/sbin/cluster/cl_event_summary.txt** file. This file continues to accumulate as **hacmp.out** cycles, and is *not* automatically truncated or replaced. Consequently, it can grow too large and crowd your **/usr** directory. You should clear event summaries periodically, using the **Remove Event Summary History** option in SMIT.

This feature is node-specific. Therefore, you cannot access one node's event summary information from another node in the cluster. Run the **View Event Summaries** option on each node for which you want to gather and display event summaries.

The event summaries display is a good way to get a quick overview of what has happened in the cluster lately. If the event summaries reveal a problem event, you will probably want to examine the source **hacmp.out** file to see full details of what happened.

Note: If you have set your formatting options for **hacmp.out** to **Default (None)**, then no event summaries will be generated. The **View Event Summaries** command will yield no results.

How Event Summary View Information Is Gathered

The **Problem Determination Tools > HACMP Log Viewing and Management > View Event Summaries** option gathers information from the **hacmp.out** log file, *not* directly from HACMP while it is running. Consequently, you can access event summary information even when HACMP is *not* running. The summary display is updated once per day with the current day's event summaries.

In addition, at the bottom of the display the resource group location and state information is shown. This information reflects output from the **clRGinfo** command.

Note that **clRGinfo** displays resource group information more quickly when the cluster is running. If the cluster is *not* running, wait a few minutes and the resource group information will eventually appear.

Viewing Event Summaries

To view a compiled list of event summaries on a node:

1. Enter `smit hacmp`
2. In SMIT, select **View Event Summaries** and press Enter. SMIT displays a list of event summaries generated on the node. SMIT will notify you if no event summaries were found.

Saving Event Summaries to a Specified File

To store the compiled list of a node's event summaries to a file:

1. Enter `smit hacmp`
2. In SMIT, select **View/Save/Remove HACMP Event Summaries**.
3. Select **Save Event Summaries to a file**.
4. Enter the path/file name where you wish to store the event summaries.

Depending on the format you select (for example .txt or .html), you can then move this file to be able to view it in a text editor or browser.

Removing Event Summaries

When you select the **Remove Event Summary History** option, HACMP deletes all event summaries compiled from **hacmp.out** files. A new list is then started.

Note: You should clear the event summary history periodically to keep the **/usr/es/sbin/cluster/cl_event_summary.txt** file from crowding your **/usr** directory.

Follow the steps below to delete the list of summaries:

1. Enter `smit hacmp`
2. In SMIT, select **View/Save/Remove HACMP Event Summaries.**
3. Select **Remove Event Summary History**. HACMP deletes all event summaries from the file.

Understanding the System Error Log

The HACMP software logs messages to the system error log whenever a daemon generates a state message.

Format of Messages in the System Error Log

The HACMP messages in the system error log follow the same format used by other AIX subsystems. You can view the messages in the system error log in short or long format.

In short format, also called summary format, each message in the **system error log** occupies a single line. The description of the fields in the short format of the **system error log**:

Error_ID	A unique error identifier.
Time stamp	The day and time on which the event occurred.
T	Error type: **permanent (P)**, **unresolved (U)**, or **temporary (T)**.
CL	Error class: **hardware (H)**, **software (S)**, or **informational (O).**
Resource_name	A text string that identifies the AIX resource or subsystem that generated the message. HACMP messages are identified by the name of their daemon.
Error_description	A text string that briefly describes the error.

In long format, a page of formatted information is displayed for each error.

Unlike the HACMP log files, the **system error log** is *not* a text file.

Using the AIX Error Report Command

The AIX **errpt** command generates an error report from entries in the system error log. For information on using this command see the **errpt** man page.

Viewing the System Error Log Using SMIT

To view the AIX **system error log**, you must use the AIX SMIT:

1. Enter `smit`
2. In SMIT, select **Problem Determination Tools** > **HACMP Log Viewing and Management > View Detailed HACMP Log Files > Scan the HACMP for AIX System Log** and press Enter.

   SMIT displays the error log.

For more information on this log file, refer to your AIX documentation.

Understanding the Cluster History Log File

The **cluster history log** file is a standard text file with the system-assigned name **/usr/es/sbin/cluster/history/cluster.*mmddyyyy***, where *mm* indicates the month, *dd* indicates the day in the month and *yyyy* indicates the year. You should decide how many of these log files you want to retain and purge the excess copies on a regular basis to conserve disk storage space. You may also decide to include the **cluster history log** file in your regular system backup procedures.

Format of Messages in the Cluster History Log File

The description of the fields in the **cluster history log** file messages:

Date and Time stamp	The date and time at which the event occurred.
Message	Text of the message.
Description	Name of the event script.

Note: This log reports specific events. Note that when resource groups are processed in parallel, certain steps previously run as separate events are now processed differently, and therefore do *not* show up as events in the cluster history log file. You should use the **hacmp.out** file, which contains greater detail on resource group activity and location, to track parallel processing activity.

Viewing the Cluster History Log File

Because the **cluster history log** file is a standard text file, you can view its contents using standard AIX file commands, such as **cat**, **more**, and **tail**. You cannot view this log file using SMIT.

Understanding the Cluster Manager Debug Log File

The **/var/hacmp/log/clstrmgr.debug.long** file contains a high level overview of the activity of the cluster manager daemon and its interaction with other components, such as the event scripts, RSCT subsystems, and the system resource controller.

The **/var/hacmp/log/clstrmgr.debug** file is a standard text file that contains the debug messages generated by the Cluster Manager. IBM Support uses this file. In terse mode, the default debug levels are recorded. In verbose mode, all debug levels are recorded.

Format of Messages in the Cluster Manager Debug Log File

The **clstrmgr.debug** log file contains time-stamped, formatted messages generated by HACMP **clstrmgrES** activity.

Viewing the Cluster Manager Debug Log File

Because the **clstrmgr.debug** log file is a standard text file, you can view its contents using standard AIX file commands, such as **cat**, **more**, and **tail**. You cannot view this log file using SMIT.

Understanding the cspoc.log.long File

The **/var/hacmp/log/cspoc.log.long** file is a standard text file that resides on the source node—the node on which the C-SPOC command is invoked. Many error messages cascade from an underlying AIX error that usually indicates the problem source and success or failure status.

Format of Messages in the cspoc.log.long File

Each **/var/hacmp/log/cspoc.log.long** entry contains a command delimiter to separate C-SPOC command output. The first line of the command's output, which contains arguments (parameters) passed to the command, follows this delimiter. Additionally, each entry contains the following information:

Date and Time stamp	The date and time the command was issued.
Node	The name of the node on which the command was executed.
Status	Text indicating the command's success or failure. Command output that reports a failure also includes the command's return code. No return code is generated for successful command completion.
Error Message	Text describing the actual error. The message is recorded in the Error message field. **Note:** Error messages generated as a result of standard C-SPOC validation are printed to **stderr** and to the **/var/hacmp/log/cspoc.log.long** file.

Viewing the cspoc.log.long File

The **/var/hacmp/log/cspoc.log.long** file is a standard text file that can be viewed in any of the following ways:

- Using standard AIX file commands, such as the **more** or **tail** commands
- Using the SMIT interface.

Using the SMIT Interface to View the cspoc.log.long File

To view the **/var/hacmp/log/cspoc.log.long** file using SMIT:

1. Enter `smit hacmp`.
2. In SMIT, select **Problem Determination Tools > HACMP Log Viewing and Management > View Detailed HACMP Log Files > Scan the C-SPOC System Log File**.

Note: Note that you can select to either *scan* the contents of the **cspoc.log.long** file as it exists, or you can *watch* an active log file as new events are appended to it in real time. Typically, you *scan* the file to try to find a problem that has already occurred; you *watch* the file while duplicating a problem to help determine its cause, or as you test a solution to a problem to determine the results.

Understanding the emuhacmp.out File

The **/var/hacmp/log/emuhacmp.out** file is a standard text file that resides on the node from which the HACMP Event Emulator was invoked. The file contains information from log files generated by the Event Emulator on all nodes in the cluster. When the emulation is complete, the information in these files is transferred to the **emuhacmp.out** file on the node from which the emulation was invoked, and all other files are deleted.

Using the EMUL_OUTPUT environment variable, you can specify another name and location for this output file. The format of the file does *not* change.

Format of Messages in the emuhacmp.out File

The entries in the **/var/hacmp/log/emuhacmp.out** file use the following format:

```
************************************************************************
*****************START OF EMULATION FOR NODE buzzcut***************
************************************************************************

Jul 21 17:17:21 EVENT START: node_down buzzcut graceful

+ [ buzzcut = buzzcut -a graceful = forced ]
+ [ EMUL = EMUL ]
+ cl_echo 3020 NOTICE >>>> The following command was not executed <<<<
\n
NOTICE >>>> The following command was not executed <<<<
+ echo /usr/es/sbin/cluster/events/utils/cl_ssa_fence down buzzcut\n
/usr/es/sbin/cluster/events/utils/cl_ssa_fence down buzzcut

+ [ 0 -ne 0 ]
+ [ EMUL = EMUL ]
+ cl_echo 3020 NOTICE >>>> The following command was not executed <<<<
\n
NOTICE >>>> The following command was not executed <<<<
```

```
+ echo /usr/es/sbin/cluster/events/utils/cl_ssa_fence down buzzcut
graceful\n
/usr/es/sbin/cluster/events/utils/cl_ssa_fence down buzzcut graceful

****************END OF EMULATION FOR NODE BUZZCUT ********************
```

The output of emulated events is presented as in the **/var/hacmp/log/hacmp.out** file described earlier in this chapter. The **/var/hacmp/log/emuhacmp.out** file also contains the following information:

Header	Each node's output begins with a header that signifies the start of the emulation and the node from which the output is received.
Notice	The **Notice** field identifies the name and path of commands or scripts that are echoed only. If the command being echoed is a customized script, such as a pre- or post-event script, the contents of the script are displayed. Syntax errors in the script are also listed.
ERROR	The **error** field contains a statement indicating the type of error and the name of the script in which the error was discovered.
Footer	Each node's output ends with a footer that signifies the end of the emulation and the node from which the output is received.

Viewing the /var/hacmp/log/emuhacmp.out File

You can view the **/var/hacmp/log/emuhacmp.out** file using standard AIX file commands. You cannot view this log file using the SMIT interface.

Collecting Cluster Log Files for Problem Reporting

If you encounter a problem with HACMP and report it to IBM support, you may be asked to collect log files pertaining to the problem. In HACMP 5.2 and up, the **Collect HACMP Log Files for Problem Reporting** SMIT panel aids in this process.

WARNING: Use this panel only if requested by the IBM support personnel. If you use this utility without direction from IBM support, be careful to fully understand the actions and the potential consequences.

To collect cluster log files for problem reporting:

1. Enter `smit hacmp`
2. In SMIT, select **Problem Determination Tools > HACMP Log Viewing and Management > Collect Log Files for Problem Reporting**.
3. Type or select values in entry fields:

Log Destination Directory	Enter a directory name where cluster logs will be collected. The default is **/tmp**.
Collection Pass Number	Select a value in this field. The default is **2** (collect). Select **1** to calculate the amount of space needed. Select **2** to collect the actual data.

Nodes to Collect Data from	Enter or select nodes from which the data will be collected. Separate node names with a comma. The default is **All** nodes.
Debug	The default is **No**. Use this option if IBM Support requests to turn on debugging.
Collect RSCT Log Files	The default is **Yes**. Skip collection of RSCT data.

Tracking Resource Group Parallel and Serial Processing in the hacmp.out File

Output to the **hacmp.out** file lets you isolate details related to a specific resource group and its resources. Based on the content of the **hacmp.out** event summaries, you can determine whether or *not* the resource groups are being processed in the expected order.

Depending on whether resource groups are processed serially or in parallel, you will see different output in the event summaries and in the log files. In HACMP, parallel processing is the default method. If you migrated the cluster from an earlier version of HACMP, serial processing is maintained.

Note: If you configured dependent resource groups and specified the serial order of processing, the rules for processing dependent resource groups override the serial order. To avoid this, the serial order of processing that you specify should *not* contradict the configured dependencies between resource groups.

This section contains detailed information on the following:

- Serial Processing Order Reflected in Event Summaries
- Parallel Processing Order Reflected in Event Summaries
- Job Types: Parallel Resource Group Processing
- Processing in Clusters with Dependent Resource Groups or Sites
- Disk Fencing with Serial or Parallel Processing.

Serial Processing Order Reflected in Event Summaries

Note: For HACMP 5.3 and up, the JOB_TYPE=NONE option is used for serial processing.

If you have defined customized serial processing lists for some of the resource groups, you can determine whether or *not* the resource groups are being processed in the expected order based on the content of the **hacmp.out** file event summaries.

The following example shows an event summary for two serially-processed resource groups named `cascrg1` and `cascrg2`:

```
              HACMP Event Summary

Event: node_ up electron

Start time: Wed May 8 11: 06: 30 2002
End time: Wed May 8 11: 07: 49 2002

Action:               Resource:               Script Name:
------------------------------------------------------------
Acquiring resource group: cascrg1 node_ up_ local
Search on: Wed. May 8. 11: 06: 33. EDT. 2002. node_ up_ local.cascrg1.
Acquiring resource: 192. 168. 41. 30 cl_ swap_ IP_ address
Search on: Wed. May. 8. 11: 06: 36. EDT. 2002. cl_ swap_ IP_ address.
192. 168.
Acquiring resource: hdisk1 cl_ disk_ available
Search on: Wed. May. 8. 11: 06: 40. EDT. 2002. cl_ disk_ available.
hdisk1. ca Resource online: hdisk1 cl_ disk_ available
Search on: Wed. May. 8. 11: 06: 42. EDT. 2002. cl_ disk_ available.
hdisk1. ca

. . .

Acquiring resource group: cascrg2 node_ up_ local
Search on: Wed. May. 8. 11: 07: 14. EDT. 2002. node_ up_ local. cascrg2.
ref
Acquiring resource: hdisk2 cl_ disk_ available
Search on: Wed. May. 8. 11: 07: 20. EDT. 2002. cl_ disk_ available.
hdisk2. ca Resource online: hdisk2 cl_ disk_ available
Search on: Wed. May. 8. 11: 07: 23. EDT. 2002. cl_ disk_ available.
hdisk2. ca
```

As shown here, each resource group appears with all of its accounted resources below it.

Parallel Processing Order Reflected in Event Summaries

The following features, listed in the **hacmp.out** file and in the event summaries, help you to follow the flow of parallel resource group processing:

- Each line in the **hacmp.out** file flow includes the name of the resource group to which it applies
- The event summary information includes details about all resource types
- Each line in the event summary indicates the related resource group.

The following example shows an event summary for resource groups named `cascrg1` and `cascrg2` that are processed in parallel:

```
              HACMP Event Summary
```

```
Event: node_ up electron

Start time: Wed May 8 11: 06: 30 2002
End time: Wed May 8 11: 07: 49 2002

Action:               Resource:               Script Name:
------------------------------------------------------------

Acquiring resource group: cascrg1 process_ resources
Search on: Wed. May. 8. 11: 06: 33. EDT. 2002. process_ resources.
cascrg1. ref
Acquiring resource group: cascrg2 process_ resources
Search on: Wed. May. 8. 11: 06: 34. EDT. 2002. process_ resources.
cascrg2. ref
Acquiring resource: 192. 168. 41. 30 cl_ swap_ IP_ address
Search on: Wed. May. 8. 11: 06: 36. EDT. 2002. cl_ swap_ IP_ address.
192. 168. 41. 30
Acquiring resource: hdisk1 cl_ disk_ available
Search on: Wed. May. 8. 11: 06: 40. EDT. 2002. cl_ disk_ available.
hdisk1. cascrg1
Acquiring resource: hdisk2 cl_ disk_ available
Search on: Wed. May. 8. 11: 06: 40. EDT. 2002. cl_ disk_ available.
hdisk2. cascrg2 Resource online: hdisk1 cl_ disk_ available
Search on: Wed. May. 8. 11: 06: 42. EDT. 2002. cl_ disk_ available.
hdisk1. cascrg1 Resource online: hdisk2 cl_ disk_ available
Search on: Wed. May. 8. 11: 06: 43. EDT. 2002. cl_ disk_ available.
hdisk2. cascrg2
```

As shown here, all processed resource groups are listed first, followed by the individual resources that are being processed.

Job Types: Parallel Resource Group Processing

The **process_resources** event script uses different JOB_TYPES that are launched during parallel processing of resource groups.

If resource group dependencies or sites are configured in the cluster, it is also useful to check the event preamble which lists the plan of action the Cluster Manager will follow to process the resource groups for a given event.

Job types are listed in the **hacmp.out** log file and help you identify the sequence of events that take place during acquisition or release of different types of resources. Depending on the cluster's resource group configuration, you may see many specific job types that take place during parallel processing of resource groups.

- There is one job type for each resource type: DISKS, FILESYSTEMS, TAKEOVER_LABELS, TAPE_RESOURCES, AIX_FAST_CONNECTIONS, APPLICATIONS, COMMUNICATION_LINKS, CONCURRENT_VOLUME_GROUPS, EXPORT_FILESYSTEMS, and MOUNT_FILESYSTEMS.
- There are also a number of job types that are used to help capitalize on the benefits of parallel processing: SETPRKEY, TELINIT, SYNC_VGS, LOGREDO, and UPDATESTATD. The related operations are now run once per event, rather than once per resource group. This is one of the primary areas of benefit from parallel resource group processing, especially for small clusters.

The following sections describe some of the most common job types in more detail and provide abstracts from the events in the **hacmp.out** log file which include these job types.

JOB_TYPE=ONLINE

In the complete phase of an acquisition event, after all resources for all resource groups have been successfully acquired, the **ONLINE** job type is run. This job ensures that all successfully acquired resource groups are set to the online state. The **RESOURCE_GROUPS** variable contains the list of all groups that were acquired.

```
:process_resources[1476] clRGPA
:clRGPA[48] [[ high = high ]]
:clRGPA[48] version= 1. 16
:clRGPA[50] usingVer= clrgpa
:clRGPA[55] clrgpa
:clRGPA[56] exit 0
:process_resources[1476] eval JOB_TYPE= ONLINE RESOURCE_GROUPS="
cascrg1 cascrg2 conc_ rg1"

:process_resources[1476] JOB_TYPE= ONLINE RESOURCE_GROUPS= cascrg1
cascrg2 conc_rg1 :process_resources[1478] RC= 0
:process_resources[1479] set +a
:process_resources[1481] [ 0 -ne 0 ]
:process_resources[1700] set_resource_group_state UP
```

JOB_TYPE= OFFLINE

In the complete phase of a release event, after all resources for all resource groups have been successfully released, the **OFFLINE** job type is run. This job ensures that all successfully released resource groups are set to the offline state. The **RESOURCE_GROUPS** variable contains the list of all groups that were released.

```
conc_rg1 :process_resources[1476] clRGPA
conc_rg1 :clRGPA[48] [[ high = high ]]
conc_rg1 :clRGPA[48] version= 1. 16
conc_rg1 :clRGPA[50] usingVer= clrgpa
conc_rg1 :clRGPA[55] clrgpa
conc_rg1 :clRGPA[56] exit 0
conc_rg1 :process_resources[1476] eval JOB_TYPE= OFFLINE
RESOURCE_GROUPS=" cascrg2 conc_ rg1"

conc_ rg1:process_resources[1476] JOB_TYPE= OFFLINE RESOURCE_GROUPS=
cascrg2 conc_rg1

conc_ rg1 :process_resources[1478] RC= 0
conc_rg1 :process_resources[1479] set +a
conc_rg1 :process_resources[1481] [ 0 -ne 0 ]
conc_rg1 :process_resources[1704] set_resource_group_state DOWN
```

JOB_TYPE=ERROR

If an error occurred during the acquisition or release of any resource, the **ERROR** job type is run. The variable **RESOURCE_GROUPS** contains the list of all groups where acquisition or release failed during the current event. These resource groups are moved into the error state. When this job is run during an acquisition event, HACMP uses the Recovery from Resource Group Acquisition Failure feature and launches an **rg_move** event for each resource group in the error state. For more information, see the Handling of Resource Group Acquisition Failures section in Appendix B: Resource Group Behavior During Cluster Events in the *Administration Guide*.

```
conc_rg1: process_resources[1476] clRGPA
conc_rg1: clRGPA[50] usingVer= clrgpa
conc_rg1: clRGPA[55] clrgpa
conc_rg1: clRGPA[56] exit 0
conc_rg1: process_resources[1476] eval JOB_ TYPE= ERROR
RESOURCE_GROUPS=" cascrg1"

conc_rg1: process_resources[1476] JOB_TYPE= ERROR RESOURCE_GROUPS=
cascrg1
conc_rg1: process_resources[1478] RC= 0
conc_rg1: process_resources[1479] set +a
conc_rg1: process_resources[1481] [ 0 -ne 0 ]
conc_rg1: process_resources[1712] set_resource_group_state ERROR
```

JOB_TYPE=NONE

After all processing is complete for the current **process_resources** script, the final job type of **NONE** is used to indicate that processing is complete and the script can return. When exiting after receiving this job, the **process_resources** script always returns `0` for success.

```
conc_rg1: process_resources[1476] clRGPA
conc_rg1: clRGPA[48] [[ high = high ]]
conc_rg1: clRGPA[48] version= 1.16
conc_rg1: clRGPA[50] usingVer= clrgpa
conc_rg1: clRGPA[55] clrgpa
conc_rg1: clRGPA[56] exit 0
conc_rg1: process_resources[1476] eval JOB_TYPE= NONE
conc_rg1: process_resources[1476] JOB_TYPE= NONE
conc_rg1: process_resources[1478] RC= 0
conc_rg1: process_resources[1479] set +a
conc_rg1: process_resources[1481] [ 0 -ne 0 ]
conc_rg1: process_resources[1721] break
conc_rg1: process_resources[1731] exit 0
```

JOB_TYPE=ACQUIRE

The **ACQUIRE** job type occurs at the beginning of any resource group acquisition event. Search **hacmp. out** for **JOB_ TYPE= ACQUIRE** and view the value of the **RESOURCE_ GROUPS** variable to see a list of which resource groups are being acquired in parallel during the event.

```
:process_resources[1476] clRGPA
:clRGPA[48] [[ high = high ]]
:clRGPA[48] version= 1. 16
:clRGPA[50] usingVer= clrgpa
:clRGPA[55] clrgpa
:clRGPA[56] exit 0
:process_resources[1476] eval JOB_TYPE= ACQUIRE RESOURCE_GROUPS="
cascrg1 cascrg2"
:process_resources[1476] JOB_TYPE= ACQUIRE RESOURCE_GROUPS= cascrg1
cascrg2
:process_resources[1478] RC= 0
:process_resources[1479] set +a
:process_resources[1481] [ 0 -ne 0 ]
:process_resources[1687] set_resource_group_state ACQUIRING
```

JOB_TYPE=RELEASE

The **RELEASE** job type occurs at the beginning of any resource group release event. Search **hacmp. out** for **JOB_ TYPE= RELEASE** and view the value of the **RESOURCE_ GROUPS** variable to see a list of which resource groups are being released in parallel during the event.

```
:process_resources[1476] clRGPA
:clRGPA[48] [[ high = high ]]
:clRGPA[48] version= 1. 16
:clRGPA[50] usingVer= clrgpa
:clRGPA[55] clrgpa
:clRGPA[56] exit 0
:process_resources[1476] eval JOB_ TYPE= RELEASE RESOURCE_ GROUPS="
cascrg1 cascrg2"
:process_resources[1476] JOB_ TYPE= RELEASE RESOURCE_ GROUPS= cascrg1
cascrg2
:process_resources[1478] RC= 0
:process_resources[1479] set +a
:process_resources[1481] [ 0 -ne 0 ]
:process_resources[1691] set_ resource_ group_ state RELEASING
```

JOB_TYPE= SSA_FENCE

The **SSA_FENCE** job type is used to handle fencing and unfencing of SSA disks. The variable **ACTION** indicates what should be done to the disks listed in the **HDISKS** variable. All resources groups (both parallel and serial) use this method for disk fencing.

```
:process_resources[1476] clRGPA FENCE
:clRGPA[48] [[ high = high ]]
:clRGPA[55] clrgpa FENCE
:clRGPA[56] exit 0
:process_resources[1476] eval JOB_TYPE= SSA_ FENCE ACTION= ACQUIRE
HDISKS=" hdisk6" RESOURCE_GROUPS=" conc_ rg1 " HOSTS=" electron"
:process_ resources[1476] JOB_TYPE= SSA_FENCE ACTION= ACQUIRE HDISKS=
hdisk6 RESOURCE_GROUPS= conc_rg1 HOSTS=electron
:process_ resources[1478] RC= 0
:process_ resources[1479] set +a
:process_ resources[1481] [ 0 -ne 0 ]
:process_ resources[1675] export GROUPNAME= conc_ rg1 conc_ rg1
:process_ resources[1676] process_ ssa_ fence ACQUIRE
```

Note: Notice that disk fencing uses the **process_resources** script, and, therefore, when disk fencing occurs, it may mislead you to assume that resource processing is taking place, when, in fact, only disk fencing is taking place. If disk fencing is enabled, you will see in the **hacmp.out** file that the disk fencing operation occurs *before* any resource group processing.

Although the **process_ resources** script handles SSA disk fencing, the resource groups are processed serially. **cl_ ssa_ fence** is called once for each resource group that requires disk fencing. The **hacmp.out** content indicates which resource group is being processed.

```
conc_ rg1: process_resources[8] export GROUPNAME
conc_ rg1: process_resources[10] get_ list_ head hdisk6
conc_ rg1: process_resources[10] read LIST_OF_HDISKS_ FOR_ RG
conc_ rg1: process_resources[11] read HDISKS
conc_ rg1: process_resources[11] get_ list_ tail hdisk6
conc_ rg1: process_resources[13] get_ list_ head electron
conc_ rg1: process_resources[13] read HOST_ FOR_ RG
conc_ rg1: process_resources[14] get_ list_ tail electron
conc_ rg1: process_resources[14] read HOSTS
conc_ rg1: process_resources[18] cl_ ssa_fence ACQUIRE electron
hdisk6
conc_ rg1: cl_ssa_fence[43] version= 1. 9. 1. 2
conc_ rg1: cl_ssa_fence[44]
conc_ rg1: cl_ssa_fence[44]
conc_ rg1: cl_ssa_fence[46] STATUS= 0
conc_ rg1: cl_ssa_fence[48] (( 3 < 3
conc_ rg1: cl_ssa_fence[56] OPERATION= ACQUIRE
```

JOB_TYPE=SERVICE_LABELS

The **SERVICE_LABELS** job type handles the acquisition or release of service labels. The variable **ACTION** indicates what should be done to the service IP labels listed in the **IP_LABELS** variable.

```
conc_ rg1: process_ resources[ 1476] clRGPA
conc_ rg1: clRGPA[ 55] clrgpa
conc_ rg1: clRGPA[ 56] exit 0
conc_ rg1: process_ resources[ 1476] eval JOB_ TYPE= SERVICE_ LABELS
ACTION= ACQUIRE IP_ LABELS=" elect_ svc0: shared_ svc1, shared_ svc2"
RESOURCE_ GROUPS=" cascrg1 rotrg1" COMMUNICATION_ LINKS=": commlink1"
conc_ rg1: process_ resources[1476] JOB_ TYPE= SERVICE_ LABELS
ACTION= ACQUIRE IP_ LABELS= elect_ svc0: shared_ svc1, shared_ svc2
RESOURCE_ GROUPS= cascrg1 rotrg1 COMMUNICATION_ LINKS=: commlink1
conc_ rg1: process_ resources[1478] RC= 0
conc_ rg1: process_ resources[1479] set +a
conc_ rg1: process_ resources[1481] [ 0 -ne 0 ]
conc_ rg1: process_ resources[ 1492] export GROUPNAME= cascrg1
```

This job type launches an **acquire_service_addr** event. Within the event, each individual service label is acquired. The content of the **hacmp.out** file indicates which resource group is being processed. Within each resource group, the event flow is the same as it is under serial processing.

```
cascrg1: acquire_service_addr[ 251] export GROUPNAME
cascrg1: acquire_service_addr[251] [[ true = true ]]
cascrg1: acquire_service_addr[254] read SERVICELABELS
cascrg1: acquire_service_addr[254] get_ list_ head electron_ svc0
cascrg1: acquire_service_addr[255] get_ list_ tail electron_ svc0
cascrg1: acquire_service_addr[255] read IP_ LABELS
cascrg1: acquire_service_addr[257] get_ list_ head
cascrg1: acquire_service_addr[257] read SNA_ CONNECTIONS
```

```
cascrg1: acquire_service_addr[258] export SNA_ CONNECTIONS
cascrg1: acquire_service_addr[259] get_ list_ tail
cascrg1: acquire_service_addr[259] read _SNA_ CONNECTIONS
cascrg1: acquire_service_addr[270] clgetif -a electron_ svc0
```

JOB_TYPE=VGS

The **VGS** job type handles the acquisition or release of volume groups. The variable **ACTION** indicates what should be done to the volume groups being processed, and the names of the volume groups are listed in the **VOLUME_GROUPS** and **CONCURRENT_VOLUME_GROUPS** variables.

```
conc_rg1 :process_resources[1476] clRGPA
conc_rg1 :clRGPA[55] clrgpa
conc_rg1 :clRGPA[56] exit 0

conc_rg1 :process_resources[1476] eval JOB_TYPE= VGS ACTION= ACQUIRE
CONCURRENT_VOLUME_GROUP=" con_vg6" VOLUME_GROUPS=""
casc_vg1: casc_vg2" RESOURCE_GROUPS=" cascrg1 cascrg2 "
EXPORT_FILESYSTEM=""

conc_rg1 :process_resources[1476] JOB_TYPE= VGS ACTION= ACQUIRE
CONCURRENT_VOLUME_GROUP= con_vg6 VOLUME_GROUPS= casc_vg1: casc_ vg2
RESOURCE_GROUPS= cascrg1 cascrg2 EXPORT_FILESYSTEM=""

conc_rg1 :process_resources[1478] RC= 0
conc_rg1 :process_resources[1481] [ 0 -ne 0 ]
conc_rg1 :process_resources[1529]
export GROUPNAME= cascrg1 cascrg2
```

This job type runs the **cl_activate_vgs** event utility script, which acquires each individual volume group. The content of the **hacmp.out** file indicates which resource group is being processed, and within each resource group, the script flow is the same as it is under serial processing.

```
cascrg1 cascrg2 :cl_activate_vgs[256] 1> /usr/ es/ sbin/ cluster/
etc/ lsvg. out. 21266 2> /tmp/ lsvg. err

cascrg1: cl_activate_vgs[260] export GROUPNAME
cascrg1: cl_activate_vgs[262] get_ list_head
casc_vg1: casc_vg2
cascrg1: cl_activate_vgs[ 62] read_LIST_OF_VOLUME_GROUPS_FOR_RG
cascrg1: cl_activate_vgs[263] get_list_tail casc_vg1: casc_vg2
cascrg1: cl_activate_vgs[263] read VOLUME_GROUPS
cascrg1: cl_activate_vgs[265] LIST_OF_VOLUME_GROUPS_ FOR_ RG=
cascrg1: cl_activate_vgs[ 270] fgrep -s -x casc_ vg1 /usr/ es/ sbin/
cluster/ etc/ lsvg. out. 21266
cascrg1: cl_activate_vgs[275] LIST_OF_VOLUME_GROUPS_FOR_RG= casc_vg1
cascrg1: cl_activate_vgs[275] [[ casc_ vg1 = ]]
```

Disk Fencing with Serial or Parallel Processing

Disk fencing with either serial or parallel processing uses the **process_resources** script with the JOB_TYPE=SSA_FENCE as described in the previous section.

Processing in Clusters with Dependent Resource Groups or Sites

Resource groups in clusters with dependent groups or sites configured are handled with dynamic event phasing. These events process one or more resource groups at a time. Multiple non-concurrent resource groups can be processed within one **rg_move** event.

If you specify serial order of processing (HACMP to use **clsetenvgrp**) and have dependent resource groups configured, make sure that the serial order does *not* contradict the dependency specified. The resource groups dependency overrides any customized serial order in the cluster.

Also, see the examples for handling resource groups with location dependencies in the Appendix: Applications and HACMP in the *Planning Guide*.

Processing Replicated Resource Groups

HACMP uses **rg_move** events for dynamic processing of replicated resources.

JOB_TYPE=SIBLINGS provides the interface variables to the HACMP/XD product in the event script's environment and prints the appropriate SIBLING variables:

SIBLING_GROUPS; (example: rg1 rg2)

SIBLING_NODES_BY_GROUP; (example: n1 : n2) Note: colon separator

SIBLING_RELEASING_GROUPS; (example: rg4 rg5)

SIBLING_RELEASING_NODES_BY_GROUP; (example: n3 : n4) Note: colon separator

SIBLING_ACQUIRING_GROUPS; (example: rg4 rg5)

SIBLING_ACQUIRING_NODES_BY_GROUP; (example: n3 : n4) Note: colon separator

These variables are used only with the **process_resource** code path. Once The Cluster Manager sends this data to the event scripts a call to **clsetrepenv** sets the environment for a specific resource group. The SIBLING variables are printed to the environment even though the local node is *not* processing any resource groups. They reflect the environment values at the peer site.

For JOB_TYPE=ACQUIRE, along with other variables that are currently set in the environment the following variables are set on each node (both in **node_up** and **rg_move** acquire):

SIBLING_GROUPS

Every resource group that has a non-ignore site policy appears in this list of group names in the HACMP event if the resource group is in either ONLINE or ONLINE_SECONDARY state on the peer site.

SIBLING_NODES_BY_GROUP

For every resource group listed in SIBLING_GROUPS, the SIBLING_NODES_BY_GROUP variable lists the node that hosts the resource group (in either ONLINE or ONLINE_SECONDARY state).

SIBLING_ACQUIRE_GROUPS

resource group's state change information.

SIBLING_ACQUIRE_NODES_BY_GROUP

resource group's state change information.

These sets of variables provide a picture of resource group actions on the peer site during the course of the local event during the acquire phase.

For JOB_TYPE=RELEASE the following variables are used (both in **node_down** and **rg_move** release):

SIBLING_GROUPS

SIBLING_NODES_BY_GROUP

SIBLING_RELEASE_GROUPS

SIBLING_RELEASE_NODES_BY_GROUP

On a per resource group basis the following variables are tracked:

SIBLING_NODES

SIBLING_NON_OWNER_NODES

SIBLING_ACQURING_GROUPS or SIBLING_RELEASING_GROUPS

SIBLING_ACQUIRING_NODES_BY_GROUP or
SIBLING_RELEASING_GROUPS_BY_NODE

Sample Event with Siblings Output to hacmp.out

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Mar 28 09:40:42 EVENT START: rg_move a2 1ACQUIRE
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
:process_resources[1952] eval JOB_TYPE=ACQUIRE RESOURCE_GROUPS="rg3"
SIBLING_GROUPS="rg1 rg3" SIBLING_NODES_BY_GROUP="b2 : b2"
SIBLING_ACQUIRING_GROUPS="" SIBLING_ACQUIRING_NODES_BY_GROUP=""
PRINCIPAL_ACTION="ACQUIRE" AUXILLIARY_ACTION="NONE"
:process_resources[1952] JOB_TYPE=ACQUIRE RESOURCE_GROUPS=rg3
SIBLING_GROUPS=rg1 rg3 SIBLING_NODES_BY_GROUP=b2 : b2
SIBLING_ACQUIRING_GROUPS= SIBLING_ACQUIRING_NODES_BY_GROUP=
PRINCIPAL_ACTION=ACQUIRE AUXILLIARY_ACTION=NONE
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
:rg_move_complete[157] eval FORCEDOWN_GROUPS="" RESOURCE_GROUPS=""
HOMELESS_GROUPS="" ERRSTATE_GROUPS="" PRINCIPAL_ACTIONS=""
ASSOCIATE_ACTIONS="" AUXILLIARY_ACTIONS="" SIBLING_GROUPS="rg1 rg3"
SIBLING_NODES_BY_GROUP="b2 : b2" SIBLING_ACQUIRING_GROUPS="" SIBLING
_ACQUIRING_NODES_BY_GROUP="" SIBLING_RELEASING_GROUPS=""
SIBLING_RELEASING_NODES_BY_GROUP=""
:rg_move_complete[157] FORCEDOWN_GROUPS= RESOURCE_GROUPS=
HOMELESS_GROUPS= ERRSTATE_GROUPS= PRINCIPAL_ACTIONS= ASSOCIATE_ACTIONS=
AUXILLIARY_ACTIONS= SIBLING_GROUPS=rg1 rg3 SIBLING_NODES_BY_GROUP=b2 :
b2 SIBLING_ACQUIRING_GROUPS= SIBLING_ACQUIRING_NODES_BY_GROUP
= SIBLING_RELEASING_GROUPS= SIBLING_RELEASING_NODES_BY_GROUP=
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
:process_resources[1952] eval JOB_TYPE=SYNC_VGS ACTION=ACQUIRE
VOLUME_GROUPS="vg3,vg3sm" RESOURCE_GROUPS="rg3 "
:process_resources[1952] JOB_TYPE=SYNC_VGS
ACTION=ACQUIRE_VOLUME_GROUPS=vg3,vg3sm RESOURCE_GROUPS=rg3
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
rg3:process_resources[1952] eval JOB_TYPE=ONLINE RESOURCE_GROUPS="rg3"
rg3:process_resources[1952] JOB_TYPE=ONLINE RESOURCE_GROUPS=rg3
rg3:process_resources[1954] RC=0
rg3:process_resources[1955] set +a
rg3:process_resources[1957] [ 0 -ne 0 ]
```

```
rg3:process_resources[2207] set_resource_group_state UP
rg3:process_resources[3] STAT=0
rg3:process_resources[6] export GROUPNAME
rg3:process_resources[7] [ UP != DOWN ]
rg3:process_resources[9] [ REAL = EMUL ]
rg3:process_resources[14] clchdaemons -d clstrmgr_scripts -t
resource_locator -n a1 -o rg3 -v UP
rg3:process_resources[15] [ 0 -ne 0 ]
rg3:process_resources[26] [ UP = ACQUIRING ]
rg3:process_resources[31] [ UP = RELEASING ]
rg3:process_resources[36] [ UP = UP ]
rg3:process_resources[38] cl_RMupdate rg_up rg3 process_resources
Reference string: Sun.Mar.27.18:02:09.EST.2005.process_resources.rg3.ref
rg3:process_resources[39] continue
rg3:process_resources[80] return 0
rg3:process_resources[1947] true
rg3:process_resources[1949] set -a
rg3:process_resources[1952] clRGPA
rg3:clRGPA[33] [[ high = high ]]
rg3:clRGPA[33] version=1.16
rg3:clRGPA[35] usingVer=clrgpa
rg3:clRGPA[40] clrgpa
rg3:clRGPA[41] exit 0
rg3:process_resources[1952] eval JOB_TYPE=NONE
rg3:process_resources[1952] JOB_TYPE=NONE
rg3:process_resources[1954] RC=0
rg3:process_resources[1955] set +a
rg3:process_resources[1957] [ 0 -ne 0 ]
rg3:process_resources[2256] break
rg3:process_resources[2267] [[ FALSE = TRUE ]]
rg3:process_resources[2273] exit 0
:rg_move_complete[346] STATUS=0
:rg_move_complete[348] exit 0
Mar 27 18:02:10 EVENT COMPLETED: rg_move_complete a1 2 0
```

Managing a Node's HACMP Log File Parameters

Each cluster node supports two log file parameters. These allow you to:

- Set the level of debug information output by the HACMP scripts. By default, HACMP sets the debug information parameter to **high**, which produces detailed output from script execution.
- Set the output format for the **hacmp.out** log file.

To change the log file parameters for a node:

1. Enter the fastpath `smit hacmp`
2. In SMIT, select **Problem Determination Tools > HACMP Log Viewing and Management > Change/Show HACMP Log File Parameters** and press Enter.
3. Select a node from the list.
4. Enter field values as follows:

Debug Level	Cluster event scripts have two levels of logging. The **low** level only logs events and errors encountered while the script executes. The **high** (default) level logs all commands performed by the script and is strongly recommended. The **high** level provides the level of script tracing needed to resolve many cluster problems.
Formatting options for hacmp.out	Select one of these: **Default (None)** (no special format), **Standard** (include search strings), **HTML (Low)** (limited HTML formatting), or **HTML (High)** (full HTML format).

5. Press Enter to add the values into the HACMP for AIX Configuration Database.
6. Return to the main HACMP menu. Select **Extended Configuration > Extended Verification and Synchronization**.

   The software checks whether cluster services are running on any cluster node. If so, there will be no option to skip verification.
7. Select the options you want to use for verification and Press Enter to synchronize the cluster configuration and node environment across the cluster. See Chapter 7: Verifying and Synchronizing a Cluster Configuration in the *Administration Guide* for complete information on this operation.

Logging for clcomd

Logging for the **clcomd** daemon to **clcomd.log** and **clcomddiag.log** is turned on by default. The information in **clcomd.log** provides information about all connections to and from the daemon, including information for the initial connections established during discovery. Because **clcomddiag.log** contains diagnostic information for the daemon, you usually do *not* use this file in troubleshooting situations.

The following example shows the type of output generated in the **clcomd.log** file. The second and third entries are generated during the discovery process.

```
Wed May 7 12:43:13 2003: Daemon was successfully started
Wed May 7 12:44:10 2003: Trying to establish connection to node
    temporarynode0000001439363040
Wed May 7 12:44:10 2003: Trying to establish connection to node
    temporarynode0000002020023310
Wed May 7 12:44:10 2003: Connection to node temporarynode0000002020023310, success,
192.0.24.4->
Wed May 7 12:44:10 2003: CONNECTION: ACCEPTED: test2: 192.0.24.4->192.0.24.4
Wed May 7 12:44:10 2003: WARNING: /usr/es/sbin/cluster/etc/rhosts permissions
    must be -rw-------
Wed May 7 12:44:10 2003: Connection to node temporarynode0000001439363040: closed
Wed May 7 12:44:10 2003: Connection to node temporarynode0000002020023310: closed
Wed May 7 12:44:10 2003: CONNECTION: CLOSED: test2: 192.0.24.4->192.0.24.4
Wed May 7 12:44:11 2003: Trying to establish connection to node test1
Wed May 7 12:44:11 2003: Connection to node test1, success, 192.0.24.4->192.0.24.5
Wed May 7 12:44:11 2003: Trying to establish connection to node test3.
```

You can view the content of the **clcomd.log** or **clcomddiag.log** file by using the AIX **vi** or **more** commands.

You can turn off logging to **clcomddiag.log** temporarily (until the next reboot, or until you enable logging for this component again) by using the AIX **tracesoff** command. To permanently stop logging to **clcomddiag.log,** start the daemon from SRC without the **-d** flag by using the following command:

```
chssys -s clcomdES -a ""
```

Redirecting HACMP Cluster Log Files

During normal operation, HACMP produces several output log files that you can use to monitor and debug your systems. You can store a cluster log in a location other than its default directory if you so choose. If you do this, keep in mind that the minimum disk space for most cluster logs is 2MB. 14MB is recommended for **hacmp.out**.

Note: Logs should be redirected to local file systems and *not* to shared or NFS file systems. Having logs on those file systems may cause problems if the file system needs to unmount during a fallover event. Redirecting logs to NFS file systems may also prevent cluster services from starting during node reintegration.

The log file redirection function does the following:

- Checks the location of the target directory to determine whether it is part of a local or remote file system.

- Performs a check to determine whether the target directory is managed by HACMP. If it is, any attempt to redirect a log file will fail.
- Checks to ensure that the target directory is specified using an absolute path (such as “/mylogdir”) as opposed to a relative path (such as “mylogdir”).

These checks decrease the possibility that the chosen file system may become unexpectedly unavailable.

Note: The target directory must have read-write access.

Chapter 3: Investigating System Components and Solving Common Problems

This chapter guides you through the steps to investigate system components, identify problems that you may encounter as you use HACMP, and offers possible solutions.

Overview

If no error messages are displayed on the console and if examining the log files proves fruitless, you next investigate each component of your HACMP environment and eliminate it as the cause of the problem. The first section of this chapter reviews methods for investigating system components, including the RSCT subsystem. It includes these sections:

- Investigating System Components
- Checking Highly Available Applications
- Checking the HACMP Layer
- Checking the Logical Volume Manager
- Checking the TCP/IP Subsystem
- Checking the AIX Operating System
- Checking Physical Networks
- Checking Disks, Disk Adapters, and Disk Heartbeating Networks
- Checking the Cluster Communications Daemon
- Checking System Hardware

The second section provides recommendations for investigating the following areas:

- HACMP Installation Issues
- HACMP Startup Issues
- Disk and File System Issues
- Network and Switch Issues
- Cluster Communications Issues
- HACMP Takeover Issues
- Client Issues
- Miscellaneous Issues

Investigating System Components

Both HACMP and AIX provide utilities you can use to determine the state of an HACMP cluster and the resources within that cluster. Using these commands, you can gather information about volume groups or networks. Your knowledge of the HACMP system is essential. You must know the characteristics of a normal cluster beforehand and be on the lookout for deviations from the norm as you examine the cluster components. Often, the surviving cluster nodes can provide an example of the correct setting for a system parameter or for other cluster configuration information.

The following sections review the HACMP cluster components that you can check and describes some useful utilities. If examining the cluster log files does *not* reveal the source of a problem, investigate each system component using a top-down strategy to move through the layers. You should investigate the components in the following order:

1. Application layer
2. HACMP layer
3. Logical Volume Manager layer
4. TCP/IP layer
5. AIX layer
6. Physical network layer
7. Physical disk layer
8. System hardware layer

The following sections describe what you should look for when examining each layer. They also briefly describe the tools you should use to examine the layers.

Checking Highly Available Applications

As a first step to finding problems affecting a cluster, check each highly available application running on the cluster. Examine any application-specific log files and perform any troubleshooting procedures recommended in the application's documentation. In addition, check the following:

- Do some simple tests; for example, for a database application try to add and delete a record.
- Use the **ps** command to check that the necessary processes are running, or to verify that the processes were stopped properly.
- Check the resources that the application expects to be present to ensure that they are available, the file systems and volume groups for example.

Checking the HACMP Layer

If checking the application layer does *not* reveal the source of a problem, check the HACMP layer. The two main areas to investigate are:

- HACMP components and required files
- Cluster topology and configuration.

The following sections describe how to investigate these problems.

Note: These steps assume that you have checked the log files and that they do *not* point to the problem.

Checking HACMP Components

An HACMP cluster is made up of several required files and daemons. The following sections describe what to check for in the HACMP layer.

Checking HACMP Required Files

Make sure that the HACMP files required for your cluster are in the proper place, have the proper permissions (readable and executable), and are *not* zero length. The HACMP files and the AIX files modified by the HACMP software are listed in the README file that accompanies the product.

Checking Cluster Services and Processes

Check the status of the following HACMP daemons:

- The Cluster Manager (**clstrmgrES**) daemon
- The Cluster Communications (**clcomdES**) daemon
- The Cluster Information Program (**clinfoES**) daemon.

When these components are *not* responding normally, determine if the daemons are active on a cluster node. Use either the options on the SMIT **System Management (C-SPOC) > Manage HACMP Services** > **Show Cluster Services** panel or the **lssrc** command.

For example, to check on the status of all daemons under the control of the SRC, enter:

```
lssrc -a | grep active
syslogd          ras              290990       active
 sendmail         mail             270484       active
 portmap          portmap          286868       active
 inetd            tcpip            295106       active
 snmpd            tcpip            303260       active
 dpid2            tcpip            299162       active
 hostmibd         tcpip            282812       active
 aixmibd          tcpip            278670       active
 biod             nfs              192646       active
 rpc.statd        nfs              254122       active
 rpc.lockd        nfs              274584       active
 qdaemon          spooler          196720       active
 writesrv         spooler          250020       active
 ctrmc            rsct             98392        active
 clcomdES         clcomdES         204920       active
 IBM.CSMAgentRM   rsct_rm          90268        active
 IBM.ServiceRM    rsct_rm          229510       active
```

```
IBM.ERRM         rsct_rm          188602       active
IBM.AuditRM      rsct_rm          151722       active
topsvcs          topsvcs          602292       active
grpsvcs          grpsvcs          569376       active
emsvcs           emsvcs           561188       active
emaixos          emsvcs           557102       active
clstrmgrES       cluster          544802       active
gsclvmd                           565356       active
IBM.HostRM       rsct_rm          442380       active
```

To check on the status of all cluster daemons under the control of the SRC, enter:

```
lssrc -g cluster
```

Note: When you use the **-g** flag with the **lssrc** command, the status information does *not* include the status of subsystems if they are inactive. If you need this information, use the **-a** flag instead. For more information on the **lssrc** command, see the man page.

To view additional information on the status of a daemon run the **clcheck_server** command. The **clcheck_server** command makes additional checks and retries beyond what is done by **lssrc** command. For more information, see the **clcheck_server** man page.

To determine whether the Cluster Manager is running, or if processes started by the Cluster Manager are currently running on a node, use the **ps** command.

For example, to determine whether the **clstrmgrES** daemon is running, enter:

```
ps -ef | grep clstrmgrES
root  18363  3346  3 11:02:05    -  10:20
/usr/es/sbin/cluster/clstrmgrES
root  19028  19559 2 16:20:04 pts/10  0:00 grep clstrmgrES
```

See the **ps** man page for more information on using this command.

Checking for Cluster Configuration Problems

For an HACMP cluster to function properly, all the nodes in the cluster must agree on the cluster topology, network configuration, and ownership and takeover of HACMP resources. This information is stored in the Configuration Database on each cluster node.

To begin checking for configuration problems, ask yourself if you (or others) have made any recent changes that may have disrupted the system. Have components been added or deleted? Has new software been loaded on the machine? Have new PTFs or application updates been performed? Has a system backup been restored? Then run verification to ensure that the proper HACMP-specific modifications to AIX software are in place and that the cluster configuration is valid.

The cluster verification utility checks many aspects of a cluster configuration and reports any inconsistencies. Using this utility, you can perform the following tasks:

- Verify that all cluster nodes contain the same cluster topology information
- Check that all network interface cards and tty lines are properly configured, and that shared disks are accessible to all nodes that can own them
- Check each cluster node to determine whether multiple RS232 non-IP networks exist on the same tty device

- Check for agreement among all nodes on the ownership of defined resources, such as file systems, log files, volume groups, disks, and application servers
- Check for invalid characters in cluster names, node names, network names, network interface names and resource group names
- Verify takeover information.

The verification utility will also print out diagnostic information about the following:

- Custom snapshot methods
- Custom verification methods
- Custom pre or post events
- Cluster log file redirection.

If you have configured Kerberos on your system, the verification utility also determines that:

- All IP labels listed in the configuration have the appropriate service principals in the **.klogin** file on each node in the cluster
- All nodes have the proper service principals
- Kerberos is installed on all nodes in the cluster
- All nodes have the same security mode setting.

From the main HACMP SMIT panel, select **Problem Determination Tools > HACMP Verification > Verify HACMP Configuration.** If you find a configuration problem, correct it, then resynchronize the cluster.

Note: Some errors require that you make changes on each cluster node. For example, a missing application start script or a volume group with autovaryon=TRUE requires a correction on each affected node. Some of these issues can be taken care of by using HACMP File Collections.

For more information about using the cluster verification utility and HACMP File Collections, see Chapter 7: Verifying and Synchronizing a Cluster Configuration in the *Administration Guide*.

Run the **/usr/es/sbin/cluster/utilities/cltopinfo** command to see a complete listing of cluster topology. In addition to running the HACMP verification process, check for recent modifications to the node configuration files.

The command **ls -lt /etc** will list all the files in the **/etc** directory and show the most recently modified files that are important to configuring AIX, such as:

- **etc/inett.conf**
- **etc/hosts**
- **etc/services**.

It is also very important to check the resource group configuration for any errors that may *not* be flagged by the verification process. For example, make sure the file systems required by the application servers are included in the resource group with the application.

Check that the nodes in each resource group are the ones intended, and that the nodes are listed in the proper order. To view the cluster resource configuration information from the main HACMP SMIT panel, select **Extended Configuration > Extended Resource Configuration > HACMP Extended Resource Group Configuration > Show All Resources by Node or Resource Group**.

You can also run the /**usr/es/sbin/cluster/utilities/clRGinfo** command to see the resource group information.

Note: If cluster configuration problems arise after running the cluster verification utility, do *not* run C-SPOC commands in this environment as they may fail to execute on cluster nodes.

Checking a Cluster Snapshot File

The HACMP cluster snapshot facility (**/usr/es/sbin/cluster/utilities/clsnapshots**) allows you to save in a file, a record all the data that defines a particular cluster configuration. It also allows you to create your own custom snapshot methods, to save additional information important to your configuration. You can use this snapshot for troubleshooting cluster problems. The default directory path for storage and retrieval of a snapshot is **/usr/es/sbin/cluster/snapshots**.

Note that you cannot use the cluster snapshot facility in a cluster that is running different versions of HACMP concurrently.

For information on how to create and apply cluster snapshots, see Chapter 18: Saving and Restoring Cluster Configurations in the *Administration Guide.*

Information Saved in a Cluster Snapshot

The primary information saved in a cluster snapshot is the data stored in the HACMP Configuration Database classes (such as HACMPcluster, HACMPnode, and HACMPnetwork). This is the information used to recreate the cluster configuration when a cluster snapshot is applied.

The cluster snapshot does *not* save any user-customized scripts, applications, or other non-HACMP configuration parameters. For example, the name of an application server and the location of its start and stop scripts are stored in the HACMPserver Configuration Database object class. However, the scripts themselves as well as any applications they may call are *not* saved.

The cluster snapshot does *not* save any device data or configuration-specific data that is outside the scope of HACMP. For instance, the facility saves the names of shared file systems and volume groups; however, other details, such as NFS options or LVM mirroring configuration are *not* saved.

If you moved resource groups using the Resource Group Management utility **clRGmove**, once you apply a snapshot, the resource groups return to behaviors specified by their default nodelists. To investigate a cluster after a snapshot has been applied, run **clRGinfo** to view the locations and states of resource groups.

In addition to this Configuration Database data, a cluster snapshot also includes output generated by various HACMP and standard AIX commands and utilities. This data includes the current state of the cluster, node, network, and network interfaces as viewed by each cluster node, as well as the state of any running HACMP daemons.

The cluster snapshot includes output from the following commands:

cllscf	**df**	**lsfs**	**netstat**
cllsnw	**exportfs**	**lslpp**	**no**
cllsif	**ifconfig**	**lslv**	**clchsyncd**
clshowres	**ls**	**lsvg**	**cltopinfo**

In HACMP 5.1 and up, by default, HACMP no longer collects cluster log files when you create the cluster snapshot, although you can still specify to do so in SMIT. Skipping the logs collection reduces the size of the snapshot and speeds up running the snapshot utility.

You can use SMIT to collect cluster log files for problem reporting. This option is available under the **Problem Determination Tools** > **HACMP Log Viewing and Management > Collect Cluster log files for Problem Reporting** SMIT menu. It is recommended to use this option only if requested by the IBM support personnel.

If you want to add commands to obtain site-specific information, create custom snapshot methods as described in the chapter on Saving and Restoring Cluster Configurations in the *Administration Guide*.

Note that you can also use the AIX **snap -e** command to collect HACMP cluster data, including the **hacmp.out** and **clstrmgr.debug** log files.

Cluster Snapshot Files

The cluster snapshot facility stores the data it saves in two separate files, the Configuration Database data file and the Cluster State Information File, each displaying information in three sections.

Configuration Database Data File (.odm)

This file contains all the data stored in the HACMP Configuration Database object classes for the cluster. This file is given a user-defined basename with the **.odm** file extension. Because the Configuration Database information must be largely the same on every cluster node, the cluster snapshot saves the values from only one node. The cluster snapshot Configuration Database data file is an ASCII text file divided into three delimited sections:

Version section	This section identifies the version of the cluster snapshot. The characters <VER identify the start of this section; the characters </VER identify the end of this section. The cluster snapshot software sets the version number.
Description section	This section contains user-defined text that describes the cluster snapshot. You can specify up to 255 characters of descriptive text. The characters <DSC identify the start of this section; the characters </DSC identify the end of this section.

ODM data section	This section contains the HACMP Configuration Database object classes in generic AIX ODM stanza format. The characters <ODM identify the start of this section; the characters </ODM identify the end of this section.

The following is an excerpt from a sample cluster snapshot Configuration Database data file showing some of the ODM stanzas that are saved:

```
<VER
1.0
</VER

<DSC
My Cluster Snapshot
</DSC

<ODM

HACMPcluster:
   id = 1106245917
   name = "HA52_TestCluster"
   nodename = "mynode"
   sec_level = "Standard"
   sec_level_msg = ""
   sec_encryption = ""
   sec_persistent = ""
   last_node_ids = ""
   highest_node_id = 0
   last_network_ids = ""
   highest_network_id = 0
   last_site_ides = ""
   highest_site_id = 0
   handle = 1
   cluster_version = 7
   reserved1 = 0
   reserved2 = 0
   wlm_subdir = ""
   settling_time = o
   rg_distribution_policy = "node"
   noautoverification = 0
   clvernodename = ""
   clverhour = 0

HACMPnode:
   name = "mynode"
   object = "VERBOSE_LOGGING"
   value = "high"
.
.
</ODM
```

Cluster State Information File (.info)

This file contains the output from standard AIX and HACMP system management commands. This file is given the same user-defined basename with the **.info** file extension. If you defined custom snapshot methods, the output from them is appended to this file. The Cluster State Information file contains three sections:

Version section	This section identifies the version of the cluster snapshot. The characters <VER identify the start of this section; the characters </VER identify the end of this section. The cluster snapshot software sets this section.
Description section	This section contains user-defined text that describes the cluster snapshot. You can specify up to 255 characters of descriptive text. The characters <DSC identify the start of this section; the characters </DSC identify the end of this section.
Command output section	This section contains the output generated by AIX and HACMP ODM commands. This section lists the commands executed and their associated output. This section is *not* delimited in any way.

Checking the Logical Volume Manager

When troubleshooting an HACMP cluster, you need to check the following LVM entities:

- Volume groups
- Physical volumes
- Logical volumes
- File systems

Checking Volume Group Definitions

Check to make sure that all shared volume groups in the cluster are active on the correct node. If a volume group is *not* active, vary it on using the appropriate command for your configuration.

In the SMIT panel **Initialization and Standard Configuration > Configure HACMP Resource Groups > Change/Show Resources for a Resource Group (standard)**, all volume groups listed in the **Volume Groups** field for a resource group should be varied on the node(s) that have the resource group online.

Using the lsvg Command to Check Volume Groups

To check for inconsistencies among volume group definitions on cluster nodes, use the **lsvg** command to display information about the volume groups defined on each node in the cluster:

```
lsvg
```

The system returns volume group information similar to the following:

```
rootvg
datavg
```

To list only the active (varied on) volume groups in the system, use the **lsvg -o** command as follows:

```
lsvg -o
```

The system returns volume group information similar to the following:

```
rootvg
```

To list all logical volumes in the volume group, and to check the volume group status and attributes, use the **lsvg -l** command and specify the volume group name as shown in the following example:

```
lsvg -l rootvg
```

Note: The volume group must be varied on to use the **lsvg-l** command.

You can also use HACMP SMIT to check for inconsistencies: **System Management (C-SPOC) > HACMP Logical Volume Management > Shared Volume Groups** option to display information about shared volume groups in your cluster.

Checking the Varyon State of a Volume Group

You may check the status of the volume group by issuing the **lsvg <*vgname*>** command. Depending on your configuration, the **lsvg** command returns the following options:

- `vg state` could be `active` (if it is active varyon), or `passive only` (if it is passive varyon).
- `vg mode` could be `concurrent` or `enhanced concurrent`.

Here is an example of **lsvg** output:

```
# lsvg myvg

VOLUME GROUP:     Volume_Group_01 VG IDENTIFIER:  0002231b00004c00000000f2801blcc3
VG STATE:         active          PP SIZE:        16 megabyte(s)
VG PERMISSION:    read/write      TOTAL PPs:      1084 (17344 megabytes)
MAX LVs:          256             FREE PPs:       977 (15632 megabytes)
LVs:              4               USED PPs:       107 (1712 megabytes)
OPEN LVs:         0               QUORUM:         2
TOTAL PVs:        2               VG DESCRIPTORS: 3
STALE PVs:        0               STALE PPs       0
ACTIVE PVs:       2               AUTO ON:        no
MAX PPs per PV    1016            MAX PVs:        32
LTG size:         128 kilobyte (s)AUTO SYNC:      no
HOT SPARE:        no
```

Using the C-SPOC Utility to Check Shared Volume Groups

To check for inconsistencies among volume group definitions on cluster nodes in a two-node C-SPOC environment:

1. Enter `smitty hacmp`
2. In SMIT, select **System Management (C-SPOC) > HACMP Logical Volume Management > Shared Volume Groups > List All Shared Volume Groups** and press Enter to accept the default (**no**).

   A list of all shared volume groups in the C-SPOC environment appears. This list also contains enhanced concurrent volume groups included as resources in non-concurrent resource groups.

You can also use the C-SPOC **cl_lsvg** command from the command line to display this information.

Checking Physical Volumes

To check for discrepancies in the physical volumes defined on each node, obtain a list of all physical volumes known to the systems and compare this list against the list of disks specified in the **Disks** field of the **Command Status** panel. Access the **Command Status** panel through the SMIT **Extended Configuration > Extended Resource Configuration > HACMP Extended Resource Group Configuration > Show All Resources by Node or Resource Group** panel.

To obtain a list of all the physical volumes known to a node and to find out the volume groups to which they belong, use the **lspv** command. If you do *not* specify the name of a volume group as an argument, the **lspv** command displays every known physical volume in the system. For example:

```
lspv
hdisk0      0000914312e971a   rootvg
hdisk1      00000132a78e213   rootvg
hdisk2      00000902a78e21a   datavg
hdisk3      00000321358e354   datavg
```

The first column of the display shows the logical name of the disk. The second column lists the physical volume identifier of the disk. The third column lists the volume group (if any) to which it belongs.

Note that on each cluster node, AIX can assign different names (hdisk numbers) to the same physical volume. To tell which names correspond to the same physical volume, compare the physical volume identifiers listed on each node.

If you specify the logical device name of a physical volume (hdisk*x*) as an argument to the **lspv** command, it displays information about the physical volume, including whether it is active (varied on). For example:

```
lspv hdisk2
PHYSICAL VOLUME:   hdisk2              VOLUME GROUP:   abalonevg
PV IDENTIFIER:     0000301919439ba5    VG IDENTIFIER: 00003019460f63c7
PV STATE:          active              VG STATE:      active/complete
STALE PARTITIONS:  0                   ALLOCATABLE:       yes
PP SIZE:           4 megabyte(s)       LOGICAL VOLUMES:   2
TOTAL PPs:         203 (812 megabytes) VG DESCRIPTORS:    2
FREE PPs:          192 (768 megabytes)
USED PPs:          11 (44 megabytes)
FREE DISTRIBUTION: 41..30..40..40..41
USED DISTRIBUTION:  00..11..00..00..00
```

If a physical volume is inactive (*not* varied on, as indicated by question marks in the **PV STATE** field), use the appropriate command for your configuration to vary on the volume group containing the physical volume. Before doing so, however, you may want to check the system error report to determine whether a disk problem exists. Enter the following command to check the system error report:

```
errpt -a|more
```

You can also use the **lsdev** command to check the availability or status of all physical volumes known to the system.

Checking Logical Volumes

To check the state of logical volumes defined on the physical volumes, use the **lspv -l** command and specify the logical name of the disk to be checked. As shown in the following example, you can use this command to determine the names of the logical volumes defined on a physical volume:

```
lspv -l hdisk2
LV NAME        LPs     PPs     DISTRIBUTION           MOUNT POINT
lv02           50      50      25..00..00..00..25     /usr
lv04           44      44      06..00..00..32..06     /clusterfs
```

Use the **lslv** ***logicalvolume*** command to display information about the state (opened or closed) of a specific logical volume, as indicated in the **LV STATE** field. For example:

```
lslv nodeAlv

LOGICAL VOLUME: nodeAlv                VOLUME GROUP:   nodeAvg
LV IDENTIFIER:  00003019460f63c7.1     PERMISSION:     read/write
VG STATE:       active/complete        LV STATE:       opened/syncd
TYPE:           jfs                    WRITE VERIFY:   off
MAX LPs:        128                    PP SIZE:        4 megabyte(s)
COPIES:         1                      SCHED POLICY:   parallel
LPs:            10                     PPs:            10
STALE PPs:      0                      BB POLICY:      relocatable
INTER-POLICY:   minimum                RELOCATABLE:    yes
INTRA-POLICY:   middle                 UPPER BOUND:    32
MOUNT POINT:    /nodeAfs               LABEL:          /nodeAfs
MIRROR WRITE CONSISTENCY: on
EACH LP COPY ON A SEPARATE PV ?: yes
```

If a logical volume state is inactive (or closed, as indicated in the **LV STATE** field), use the appropriate command for your configuration to vary on the volume group containing the logical volume.

Using the C-SPOC Utility to Check Shared Logical Volumes

To check the state of shared logical volumes on cluster nodes:

In SMIT select **System Management (C-SPOC) > HACMP Logical Volume Management > Shared Logical Volumes > List All Shared Logical Volumes by Volume Group**. A list of all shared logical volumes appears.

You can also use the C-SPOC **cl_lslv** command from the command line to display this information.

Checking File Systems

Check to see if the necessary file systems are mounted and where they are mounted. Compare this information against the HACMP definitions for any differences. Check the permissions of the file systems and the amount of space available on a file system.

Use the following commands to obtain this information about file systems:

- The **mount** command
- The **df** command
- The **lsfs** command.

Use the **cl_lsfs** command to list file system information when running the C-SPOC utility.

Obtaining a List of File Systems

Use the **mount** command to list all the file systems, both JFS and NFS, currently mounted on a system and their mount points. For example:

```
mount

node mounted      mounted over    vfsdate             options
-------------------------------------------------------------------------
      /dev/hd4    /               jfsOct 06 09:48     rw,log=/dev/hd8
      /dev/hd2    /usr            jfsOct 06 09:48     rw,log=/dev/hd8
      /dev/hd9var /var            jfsOct 06 09:48     rw,log=/dev/hd8
      /dev/hd3    /tmp            jfsOct 06 09:49     rw,log=/dev/hd8
      /dev/hd1    /home           jfsOct 06 09:50     rw,log=/dev/hd8
pearl /home       /home           nfsOct 07 09:59     rw,soft,bg,intr
jade  /usr/local  /usr/local      nfsOct 07 09:59     rw,soft,bg,intr
```

Determine whether and where the file system is mounted, then compare this information against the HACMP definitions to note any differences.

Checking Available File System Space

To see the space available on a file system, use the **df** command. For example:

```
df

File System    Total KB     free %used   iused %iused Mounted on
/dev/hd4         12288     5308   56%     896    21% /
/dev/hd2        413696    26768   93%   19179    18% /usr
/dev/hd9var       8192     3736   54%     115     5% /var
/dev/hd3          8192     7576    7%      72     3% /tmp
/dev/hd1          4096     3932    4%      17     1% /home
/dev/crab1lv      8192     7904    3%      17     0% /crab1fs
/dev/crab3lv     12288    11744    4%      16     0% /crab3fs
/dev/crab4lv     16384    15156    7%      17     0% /crab4fs
/dev/crablv       4096     3252   20%      17     1% /crabfs
```

Check the **%used** column for file systems that are using more than 90% of their available space. Then check the **free** column to determine the exact amount of free space left.

Checking Mount Points, Permissions, and File System Information

Use the **lsfs** command to display information about mount points, permissions, file system size and so on. For example:

```
lsfs

Name           Nodename        Mount Pt VFS    Size    Options Auto
/dev/hd4       --              /        jfs    24576   --      yes
/dev/hd1       --              /home    jfs    8192    --      yes
/dev/hd2       --              /usr     jfs    827392  --      yes
/dev/hd9var    --              /var     jfs    16384   --      yes
/dev/hd3       --              /tmp     jfs    16384   --      yes
/dev/hd7       --              /mnt     jfs    --      --      no
/dev/hd5       --              /blv     jfs    --      --      no
/dev/crab1lv   --              /crab1fs jfs    16384   rw      no
/dev/crab3lv   --              /crab3fs jfs    24576   rw      no
/dev/crab4lv   --              /crab4fs jfs    32768   rw      no
/dev/crablv    --              /crabfs  jfs    8192    rw      no
```

Important: For file systems to be NFS exported, be sure to verify that logical volume names for these file systems are consistent throughout the cluster.

Using the C-SPOC Utility to Check Shared File Systems

To check to see whether the necessary shared file systems are mounted and where they are mounted on cluster nodes in a two-node C-SPOC environment:

In SMIT select **System Management (C-SPOC) > HACMP Logical Volume Management > Shared Filesystems**. Select from either **Journaled Filesystems > List All Shared Filesystems** or **Enhanced Journaled Filesystems > List All Shared Filesystems** to display a list of shared file systems.

You can also use the C-SPOC **cl_lsfs** command from the command line to display this information.

Checking the Automount Attribute of File Systems

At boot time, AIX attempts to check all the file systems listed in **/etc/filesystems** with the **check=true** attribute by running the **fsck** command. If AIX cannot check a file system, it reports the following error:

```
Filesystem helper: 0506-519 Device open failed
```

For file systems controlled by HACMP, this error message typically does *not* indicate a problem. The file system check fails because the volume group on which the file system is defined is *not* varied on at boot time.

To avoid generating this message, edit the **/etc/filesystems** file to ensure that the stanzas for the shared file systems do *not* include the **check=true** attribute.

Checking the TCP/IP Subsystem

To investigate the TCP/IP subsystem, use the following AIX commands:

- Use the **netstat** command to make sure that the network interfaces are initialized and that a communication path exists between the local node and the target node.
- Use the **ping** command to check the point-to-point connectivity between nodes.
- Use the **ifconfig** command on all network interfaces to detect bad IP addresses, incorrect subnet masks, and improper broadcast addresses.
- Scan the **/var/hacmp/log/hacmp.out** file to confirm that the **/etc/rc.net** script has run successfully. Look for a zero exit status.
- If IP address takeover is enabled, confirm that the **/etc/rc.net** script has run and that the service interface is on its service address and *not* on its base (boot) address.
- Use the **lssrc -g tcpip** command to make sure that the **inetd** daemon is running.
- Use the **lssrc -g portmap** command to make sure that the **portmapper** daemon is running.
- Use the **arp** command to make sure that the cluster nodes are *not* using the same IP or hardware address.
- Use the **netstat** command to:
  - Show the status of the network interfaces defined for a node.
  - Determine whether a route from the local node to the target node is defined.

    The **netstat -in** command displays a list of all initialized interfaces for the node, along with the network to which that interface connects and its IP address. You can use this command to determine whether the service and standby interfaces are on separate subnets. The subnets are displayed in the **Network** column.

```
netstat -in

Name    Mtu    Network       Address         Ipkts   Ierrs  Opkts   Oerrs   Coll
lo0    1536    <Link>                        18406     0    18406     0      0
lo0    1536    127           127.0.0.1       18406     0    18406     0      0
en1    1500    <Link>                        1111626   0    58643     0      0
en1    1500    100.100.86.   100.100.86.136  1111626   0    58643     0      0
en0    1500    <Link>                        943656    0    52208     0      0
en0    1500    100.100.83.   100.100.83.136  943656    0    52208     0      0
tr1    1492    <Link>                        1879      0    1656      0      0
tr1    1492    100.100.84.   100.100.84.136  1879      0    1656      0      0
```

Look at the first, third, and fourth columns of the output. The **Name** column lists all the interfaces defined and available on this node. Note that an asterisk preceding a name indicates the interface is down (*not* ready for use). The **Network** column identifies the network to which the interface is connected (its subnet). The **Address** column identifies the IP address assigned to the node.

The **netstat -rn** command indicates whether a route to the target node is defined. To see all the defined routes, enter:

```
netstat -rn
```

Information similar to that shown in the following example is displayed:

```
Routing tables
Destination      Gateway            Flags  Refcnt Use       Interface
Netmasks:
(root node)
(0)0
(0)0 ff00 0
(0)0 ffff 0
(0)0 ffff ff80 0
(0)0 70 204 1 0
(root node)Route Tree for Protocol Family 2:
(root node)
127              127.0.0.1          U           3     1436  lo0
127.0.0.1        127.0.0.1          UH          0      456  lo0
100.100.83.128   100.100.83.136     U           6    18243  en0
100.100.84.128   100.100.84.136     U           1     1718  tr1
100.100.85.128   100.100.85.136     U           2     1721  tr0
100.100.86.128   100.100.86.136     U           8    21648  en1
100.100.100.128  100.100.100.136    U           0       39  en0
(root node)Route Tree for Protocol Family 6:
(root node)
(root node)
```

To test for a specific route to a network (for example 100.100.83), enter:

```
netstat -nr | grep '100\.100\.83'

100.100.83.128   100.100.83.136     U           6    18243  en0
```

The same test, run on a system that does *not* have this route in its routing table, returns no response. If the service and standby interfaces are separated by a bridge, router, or hub and you experience problems communicating with network devices, the devices may *not* be set to handle two network segments as one physical network. Try testing the devices independent of the configuration, or contact your system administrator for assistance.

Note that if you have only one interface active on a network, the Cluster Manager will *not* generate a failure event for that interface. For more information, see the section on network interface events in the *Planning Guide*.

See the **netstat** man page for more information on using this command.

Checking Point-to-Point Connectivity

The **ping** command tests the point-to-point connectivity between two nodes in a cluster. Use the **ping** command to determine whether the target node is attached to the network and whether the network connections between the nodes are reliable. Be sure to test all TCP/IP interfaces configured on the nodes (service and standby).

For example, to test the connection from a local node to a remote node named *nodeA* enter:

```
/etc/ping nodeA

PING testcluster.nodeA.com: (100.100.81.141): 56 data bytes
64 bytes from 100.100.81.141: icmp_seq=0 ttl=255 time=2 ms
64 bytes from 100.100.81.141: icmp_seq=1 ttl=255 time=1 ms
64 bytes from 100.100.81.141: icmp_seq=2 ttl=255 time=2 ms
64 bytes from 100.100.81.141: icmp_seq=3 ttl=255 time=2 ms
```

Type `Control-C` to end the display of packets. The following statistics appear:

```
----testcluster.nodeA.com PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1/1/2 ms
```

The **ping** command sends packets to the specified node, requesting a response. If a correct response arrives, **ping** prints a message similar to the output shown above indicating no lost packets. This indicates a valid connection between the nodes.

If the **ping** command hangs, it indicates that there is no valid path between the node issuing the **ping** command and the node you are trying to reach. It could also indicate that required TCP/IP daemons are *not* running. Check the physical connection between the two nodes. Use the **ifconfig** and **netstat** commands to check the configuration. A “bad value” message indicates problems with the IP addresses or subnet definitions.

Note that if “`DUP!`” appears at the end of the **ping** response, it means the **ping** command has received multiple responses for the same address. This response typically occurs when network interfaces have been misconfigured, or when a cluster event fails during IP address takeover. Check the configuration of all interfaces on the subnet to verify that there is only one interface per address. For more information, see the **ping** man page.

In addition, you can assign a *persistent node IP label* to a cluster network on a node. When for administrative purposes you wish to reach a specific node in the cluster using the **ping** or **telnet** commands without worrying whether an service IP label you are using belongs to any of the resource groups present on that node, it is convenient to use a persistent node IP label defined on that node.

For more information on how to assign persistent Node IP labels on the network on the nodes in your cluster, see the *Planning Guide* and Chapter 4: Configuring HACMP Cluster Topology and Resources (Extended) in the *Administration Guide*.

Checking the IP Address and Netmask

Use the **ifconfig** command to confirm that the IP address and netmask are correct. Invoke **ifconfig** with the name of the network interface that you want to examine. For example, to check the first Ethernet interface, enter:

```
ifconfig en0

en0: flags=2000063<UP,BROADCAST,NOTRAILERS,RUNNING,NOECHO>
       inet 100.100.83.136 netmask 0xffffff00 broadcast 100.100.83.255
```

If the specified interface does *not* exist, **ifconfig** replies:

```
No such device
```

The **ifconfig** command displays two lines of output. The first line shows the interface's name and characteristics. Check for these characteristics:

UP — The interface is ready for use. If the interface is down, use the **ifconfig** command to initialize it. For example:

```
ifconfig en0 up
```

If the interface does *not* come up, replace the interface cable and try again. If it still fails, use the **diag** command to check the device.

RUNNING — The interface is working. If the interface is *not* running, the driver for this interface may *not* be properly installed, or the interface is *not* properly configured. Review all the steps necessary to install this interface, looking for errors or missed steps.

The second line of output shows the IP address and the subnet mask (written in hexadecimal). Check these fields to make sure the network interface is properly configured.

See the **ifconfig** man page for more information.

Using the arp Command

Use the **arp** command to view what is currently held to be the IP addresses associated with nodes listed in a host's arp cache. For example:

```
arp -a

flounder (100.50.81.133) at 8:0:4c:0:12:34 [ethernet]
cod (100.50.81.195) at 8:0:5a:7a:2c:85 [ethernet]
seahorse (100.50.161.6) at 42:c:2:4:0:0 [token ring]
pollock (100.50.81.147) at 10:0:5a:5c:36:b9 [ethernet]
```

This output shows what the host node currently believes to be the IP and MAC addresses for nodes flounder, cod, seahorse and pollock. (If IP address takeover occurs without Hardware Address Takeover, the MAC address associated with the IP address in the host's arp cache may become outdated. You can correct this situation by refreshing the host's arp cache.)

See the **arp** man page for more information.

Checking Heartbeating over IP Aliases

The **hacmp.out** file shows when a heartbeating over IP Aliases addresses is removed from an interface and when it is added to the interface again during an adapter_swap.

Use the following to check the configuration for heartbeating over IP Aliases:

- **netstat -n** shows the aliases
- **clstrmgr.debug** shows an IP Alias address when it is mapped to an interface.

Checking ATM Classic IP Hardware Addresses

For Classic IP interfaces, the **arp** command is particularly useful to diagnose errors. It can be used to verify the functionality of the ATM network on the ATM protocol layer, and to verify the registration of each Classic IP client with its server.

Example 1

The following **arp** command yields the output below:

```
arp -t atm -a
SVC - at0 on device atm2
========================
at0(10.50.111.4) 39.99.99.99.99.99.99.0.0.99.99.1.1.8.0.5a.99.a6.9b.0
IP Addr       VPI:VCI Handle ATM Address
stby_1A(10.50.111.2)
      0:110 21 39.99.99.99.99.99.99.0.0.99.99.1.1.8.0.5a.99.82.48.7
server_10_50_111(10.50.111.99)
      0:103 14 39.99.99.99.99.99.99.0.0.99.99.1.1.88.88.88.88.a.11.0
stby_1C(10.50.111.6)
      0:372   11  39.99.99.99.99.99.99.0.0.99.99.1.1.8.0.5a.99.98.fc.0

SVC - at2 on device atm1
========================
at2(10.50.110.4) 39.99.99.99.99.99.99.0.0.99.99.1.1.8.0.5a.99.83.63.2
IP Addr       VPI:VCI Handle ATM Address
boot_1A(10.50.110.2)
      0:175   37 39.99.99.99.99.99.99.0.0.99.99.1.1.8.0.5a.99.9e.2d.2
server_10_50_110(10.50.110.99)
      0:172   34  39.99.99.99.99.99.99.0.0.99.99.1.1.88.88.88.88.a.10.0
boot_1C(10.50.110.6)
      0:633   20 39.99.99.99.99.99.99.0.0.99.99.1.1.8.0.5a.99.99.c1.3
```

The ATM devices atm1, and atm2, have connected to the ATM switch, and retrieved its address, 39.99.99.99.99.99.99.0.0.99.99.1.1. This address appears in the first 13 bytes of the two clients, at0, and at2. The clients have successfully registered with their corresponding Classic IP server - server_10_50_111 for at0 and server_10_50_110 for at2. The two clients are able to communicate with other clients on the same subnet. (The clients for at0, for example, are stby_1A, and stby_1C.)

Example 2

If the connection between an ATM device and the switch is *not* functional on the ATM layer, the output of the **arp** command looks as follows:

```
arp -t atm -a

SVC - at0 on device atm2
========================
at0(10.50.111.4) 8.0.5a.99.a6.9b.0.0.0.0.0.0.0.0.0.0.0.0.0.0
```

Here the MAC address of ATM device atm2, 8.0.5a.99.a6.9b, appears as the first six bytes of the ATM address for interface at0. The ATM device atm2 has *not* registered with the switch, since the switch address does not appear as the first part of the ATM address of at0.

Checking the AIX Operating System

To view hardware and software errors that may affect the cluster, use the **errpt** command. Be on the lookout for disk and network error messages, especially permanent ones, which indicate real failures. See the **errpt** man page for more information.

Checking Physical Networks

Checkpoints for investigating physical connections include:

- Check the serial line between each pair of nodes.
- If you are using Ethernet:
  - Use the **diag** command to verify that the network interface card is good.
  - Ethernet adapters for the IBM System p™ can be used either with the transceiver that is on the card or with an external transceiver. There is a jumper on the NIC to specify which you are using. Verify that your jumper is set correctly.
  - Make sure that hub lights are on for every connected cable.
- If you are using Token-Ring:
  - Use the **diag** command to verify that the NIC and cables are good.
  - Make sure that all the nodes in the cluster are on the same ring.
  - Make sure that the ringspeed is set to the same value for all NICs.

To review HACMP network requirements, see Chapter 3: Planning Cluster Network Connectivity in the *Planning Guide.*

Checking Disks, Disk Adapters, and Disk Heartbeating Networks

Use the **diag** command to verify that the adapter card is functioning properly. If problems arise, be sure to check the jumpers, cables, and terminators along the SCSI bus.

For SCSI disks, including IBM SCSI disks and arrays, make sure that each array controller, adapter, and physical disk on the SCSI bus has a unique SCSI ID. Each SCSI ID on the bus must be an integer value from 0 through 15, although some SCSI adapters may have limitations on the SCSI ID that can be set. See the device documentation for information about any device-specific limitations. A common configuration is to set the SCSI ID of the adapters on the nodes to be higher than the SCSI IDs of the shared devices. Devices with higher IDs take precedence in SCSI bus contention.

For example, if the standard SCSI adapters use IDs 5 and 6, assign values from 0 through 4 to the other devices on the bus. You may want to set the SCSI IDs of the adapters to 5 and 6 to avoid a possible conflict when booting one of the systems in service mode from a **mksysb** tape of other boot devices, since this will always use an ID of 7 as the default.

If the SCSI adapters use IDs of 14 and 15, assign values from 3 through 13 to the other devices on the bus. Refer to your worksheet for the values previously assigned to the adapters.

You can check the SCSI IDs of adapters and disks using either the **lsattr** or **lsdev** command. For example, to determine the SCSI ID of the adapter *scsi1* (SCSI-3), use the following **lsattr** command and specify the logical name of the adapter as an argument:

```
lsattr -E -l scsi1 | grep id
```

Do *not* use wildcard characters or full pathnames on the command line for the device name designation.

Important: If you restore a backup of your cluster configuration onto an existing system, be sure to recheck or reset the SCSI IDs to avoid possible SCSI ID conflicts on the shared bus. Restoring a system backup causes adapter SCSI IDs to be reset to the default SCSI ID of 7.

If you note a SCSI ID conflict, see the *Planning Guide* for information about setting the SCSI IDs on disks and disk adapters.

To determine the SCSI ID of a disk, enter:

```
lsdev -Cc disk -H
```

Recovering from PCI Hot Plug NIC Failure

If an unrecoverable error causes a PCI hot-replacement process to fail, you may be left in a state where your NIC is unconfigured and still in maintenance mode. The PCI slot holding the card and/or the new card may be damaged at this point. User intervention is required to get the node back in fully working order.

For more information, refer to your hardware manuals or search for information about devices on IBM's website.

Checking Disk Heartbeating Networks

Cluster verification confirms whether a disk heartbeating network is correctly configured. RSCT logs provide information for disk heartbeating networks that is similar for information for other types of networks.

Use the following commands to test connectivity for a disk heartbeating network:

- **dhb_read** tests connectivity for a disk heartbeating network.

  For information about **dhb_read**, see the RSCT Command for Testing Disk Heartbeating section in Appendix C: HACMP for AIX Commands in the *Administration Guide*.
- **clip_config** provides information about devices discovered for disk heartbeating.
- **lssrc -ls topsvcs** shows network activity.

Testing a Disk Heartbeating Network

The first step in troubleshooting a disk heartbeating network is to test the connections. For RS232 networks, the disk heartbeating network cannot be tested while the network is active.

To use **dhb_read** to test a disk heartbeating connection:

1. Set one node to run the command in data mode:

   `dhb_read -p` *hdisk#* `-r`

   where *hdisk#* identifies the hdisk in the network, such as hdisk1.

2. Set the other node to run the command in transmit mode:

   `dhb_read -p` *hdisk#* `-t`

   where *hdisk#* identifies the hdisk in the network, such as hdisk1.

   The *hdisk#* is the same on both nodes.

   The following message indicates that the communication path is operational:

   ```
   Link operating normally.
   ```

If a device that is expected to appear in a picklist does *not*, view the **clip_config** file to see what information was discovered.

```
$ cat /usr/es/sbin/cluster/etc/config/clip_config | grep diskhb
nodeA:15#Serial#(none)#0#/0#0##0#0.0.0.0#hdisk1#hdisk1#
DE:AD:BE:EF#(none)##diskhb#public#0#0002409f07346b43
nodeB:15#Serial#(none)#0#/0#0##0#0.0.0.0#hdisk1#hdisk1#
DE:AD:BE:EF#(none)##diskhb#public#0#0002409f07346b43
```

Disk Heartbeating Networks and Network Failure Detection

Disk heartbeating networks are identical to other non-IP based networks in terms of the operation of the failure detection rate. However, there is a subtle difference that affects the state of the network endpoints and the events run.

Disk heartbeating networks work by exchanging heartbeat messages on a reserved portion of a shared disk. As long as the node can access the disk, the network endpoint will be considered up, even if heartbeat messages are *not* being sent between nodes. The disk heartbeating network itself will still be considered down.

All other non-IP networks mark the network and both endpoints as down when either endpoint fails. This difference makes it easier to diagnose problems with disk heartbeating networks: If the problem is in the connection of just one node with the shared disk only that part of the network will be marked as down.

Disk Heartbeating Networks and Fast Node Failure Detection

HACMP 5.4.1 provides a method to reduce the time it takes for a node failure to be realized throughout the cluster, while reliably detecting node failures.

HACMP 5.4.1 uses disk heartbeating to put a departing message on a shared disk so its neighbor(s) will be immediately aware of the node failure (without waiting for missed heartbeats). Topology Services will then distribute the information about the node failure throughout the cluster and then each Topology Services daemon sends a **node_down** event to any concerned client.

For more information see the section Decreasing Node Fallover Time in Chapter 3: Planning Cluster Network Connectivity in the *Planning Guide*.

Disk Heartbeating Networks and Failed Disk Enclosures

In addition to providing a non-IP network to help ensure high availability, you can use disk heartbeating networks to detect failure of a disk enclosure (cabinet). To use this function, configure a disk heartbeating network for at least one disk in each disk enclosure.

To configure a disk heartbeating network to detect a failure of a disk enclosure:

1. Configure a disk heartbeating network for a disk in the specified enclosure. For information about configuring a disk heartbeating network, see the section Configuring Heartbeating over Disk in the *Administration Guide*.
2. Create a pre- or post-event, or a notification method, to determine the action to be taken in response to a failure of the disk heartbeating network. (A failure of the disk enclosure would be seen as a failure of the disk heartbeating network.)

Checking the Cluster Communications Daemon

In some cases, if you change or remove IP addresses in the AIX adapter configuration, and this takes place *after* the cluster has been synchronized, the Cluster Communications daemon cannot validate these addresses against the **/usr/es/sbin/cluster/etc/rhosts** file or against the entries in the HACMP's Configuration Database, and HACMP issues an error.

Or, you may obtain an error during the cluster synchronization.

In this case, you must update the information that is saved in the **/usr/es/sbin/cluster/etc/rhosts** file on all cluster nodes, and refresh **clcomd** to make it aware of the changes. When you synchronize and verify the cluster again, **clcomd** starts using IP addresses added to HACMP Configuration Database.

To refresh the Cluster Communications daemon, use:

```
refresh -s clcomdES
```

Also, configure the **/usr/es/sbin/cluster/etc/rhosts** file to contain all the addresses currently used by HACMP for inter-node communication, and then copy this file to all cluster nodes.

For troubleshooting other related problems, also see Cluster Communications Issues in this chapter.

Checking System Hardware

Check the power supplies and LED displays to see if any error codes are displayed. Run the AIX **diag** command to test the system unit.

Without an argument, **diag** runs as a menu-driven program. You can also run **diag** on a specific piece of hardware. For example:

```
diag -d hdisk0 -c

Starting diagnostics.
Ending diagnostics.
```

This output indicates that hdisk0 is okay.

HACMP Installation Issues

The following potential installation issues are described here:

- Cannot Find File System at Boot Time
- cl_convert Does Not Run Due to Failed Installation
- Configuration Files Could Not Be Merged During Installation.

Cannot Find File System at Boot Time

Problem

At boot-time, AIX tries to check, by running the **fsck** command, all the file systems listed in **/etc/filesystems** with the **check=true** attribute. If it cannot check a file system. AIX reports the following error:

```
+-----------------------------------------------------------+
     Filesystem Helper: 0506-519 Device open failed
+-----------------------------------------------------------+
```

Solution

For file systems controlled by HACMP, this error typically does *not* indicate a problem. The file system check failed because the volume group on which the file system is defined is *not* varied on at boot-time. To prevent the generation of this message, edit the **/etc/filesystems** file to ensure that the stanzas for the shared file systems do *not* include the **check=true** attribute.

cl_convert Does Not Run Due to Failed Installation

Problem

When you install HACMP, **cl_convert** is run automatically. The software checks for an existing HACMP configuration and attempts to convert that configuration to the format used by the version of the software bring installed. However, if installation fails, **cl_convert** will fail to run as a result. Therefore, conversion from the Configuration Database of a previous HACMP version to the Configuration Database of the current version will also fail.

Solution

Run **cl_convert** from the command line. To gauge conversion success, refer to the **clconvert.log** file, which logs conversion progress.

Root user privilege is required to run **cl_convert**.

WARNING: Before converting to HACMP 5.4.1, be sure that your ODMDIR environment variable is set to **/etc/es/objrepos**.

For information on **cl_convert** flags, refer to the **cl_convert** man page.

Configuration Files Could Not Be Merged During Installation

Problem

During the installation of HACMP client software, the following message appears:

```
  +-------------------------------------------------------+
                Post-installation Processing...
  +-------------------------------------------------------+
Some configuration files could not be automatically merged into the
system during the installation. The previous versions of these files
have been saved in a configuration directory as listed below. Compare
the saved files and the newly installed files to determine if you need
to recover configuration data. Consult product documentation to
determine how to merge the data.
Configuration files, which were saved in /usr/lpp/save.config:
/usr/es/sbin/cluster/utilities/clexit.rc
```

Solution

As part of the HACMP installation process, copies of HACMP files that could potentially contain site-specific modifications are saved in the **/usr/lpp/save.config** directory before they are overwritten. As the message states, you must merge site-specific configuration information into the newly installed files.

HACMP Startup Issues

The following potential HACMP startup issues are described here:

ODMPATH Environment Variable Not Set Correctly

Problem

Queried object *not* found.

Solution

HACMP has a dependency on the location of certain ODM repositories to store configuration data. The ODMPATH environment variable allows ODM commands and subroutines to query locations other than the default location if the queried object does *not* reside in the default location. You can set this variable, but it must include the default location, **/etc/objrepos**, or the integrity of configuration information may be lost.

clinfo Daemon Exits after Starting

Problem

The "smux-connect" error occurs after starting the **clinfoES** daemon with the **-a** option. Another process is using port 162 to receive traps.

Solution

Check to see if another process, such as the **trapgend** smux subagent of NetView for AIX or the System Monitor for AIX **sysmond** daemon, is using port 162. If so, restart **clinfoES** without the **-a** option and configure NetView for AIX to receive the SNMP traps. Note that you will *not* experience this error if **clinfoES** is started in its normal way using the **startsrc** command.

Node Powers Down; Cluster Manager Will Not Start

Problem

The node powers itself off or appears to hang after starting the Cluster Manager. The configuration information does *not* appear to be identical on all nodes, causing the **clexit.rc** script to issue a **halt -q** to the system.

Solution

Use the cluster verification utility to uncover discrepancies in cluster configuration information on all cluster nodes.

Correct any configuration errors uncovered by the cluster verification utility. Make the necessary changes using the HACMP Initialization and Standard Configuration or Extended Configuration SMIT panels. After correcting the problem, select the **Verify and Synchronize HACMP Configuration** option to synchronize the cluster resources configuration across all nodes. Then select the **Start Cluster Services** option from the **System Management (C-SPOC) > Manage HACMP Services** SMIT panel to start the Cluster Manager.

The Cluster Manager should *not* exit if the configuration has passed cluster verification. If it does exit, use the AIX **snap -e** command to collect HACMP cluster data, including the log files and open a Program Management Report (PMR) requesting performance assistance.

For more information about the **snap -e** command, see the section Using the AIX Data Collection Utility, in Chapter 1: Troubleshooting HACMP Clusters.

You can modify the file **/etc/cluster/hacmp.term** to change the default action after an abnormal exit. The **clexit.rc** script checks for the presence of this file, and if you have made it executable, the instructions there will be followed instead of the automatic halt called by **clexit.rc**. Please read the caveats contained in the **/etc/cluster/hacmp.term** file, before making any modifications. For more information, see the section Abnormal Termination of a Cluster Daemon in the *Administration Guide*.

configchk Command Returns an Unknown Host Message

Problem

The **/etc/hosts** file on each cluster node does *not* contain the IP labels of other nodes in the cluster. For example, in a four-node cluster, Node A, Node B, and Node C's **/etc/hosts** files do *not* contain the IP labels of the other cluster nodes.

If this situation occurs, the **configchk** command returns the following message to the console:

```
"your hostname not known," "Cannot access node x."
```

which indicates that the **/etc/hosts** file on Node x does *not* contain an entry for your node.

Solution

Before starting the HACMP software, ensure that the **/etc/hosts** file on each node includes the service and boot IP labels of each cluster node.

Cluster Manager Hangs during Reconfiguration

Problem

The Cluster Manager hangs during reconfiguration and generates messages similar to the following:

```
The cluster has been in reconfiguration too long;Something may be wrong.
```

An event script has failed.

Solution

Determine why the script failed by examining the **/var/hacmp/log/hacmp.out** file to see what process exited with a non-zero status. The error messages in the **/var/hacmp/adm/cluster.log** file may also be helpful. Fix the problem identified in the log file. Then run the **clruncmd** command either at the command line, or by using the SMIT **Problem Determination Tools > Recover From HACMP Script Failure** panel. The **clruncmd** command signals the Cluster Manager to resume cluster processing.

clcomdES and clstrmgrES Fail to Start on Newly installed AIX Nodes

Problem

On newly installed AIX nodes, **clcomdES** and **clstrmgrES** fail to start.

Solution

Manually indicate to the system console (for the AIX installation assistant) that the AIX installation is finished.

This problem usually occurs on newly installed AIX nodes; at the first boot AIX runs the installation assistant from **/etc/inittab** and does *not* proceed with other entries in this file. AIX installation assistant waits for your input on system console. AIX will run the installation assistant on every subsequent boot, until you indicate that installation is finished. Once you do so, the system will proceed to start the cluster communications daemon (**clcomdES**) and the Cluster Manager daemon (**clstrmgr**).

Pre- or Post-Event Does Not Exist on a Node after Upgrade

Problem

The cluster verification utility indicates that a pre- or post-event does *not* exist on a node after upgrading to a new version of the HACMP software.

Solution

Ensure that a script by the defined name exists and is executable on all cluster nodes.

Each node must contain a script associated with the defined pre- or post-event. While the contents of the script do *not* have to be the same on each node, the name of the script must be consistent across the cluster. If no action is desired on a particular node, a *no-op* script with the same event-script name should be placed on nodes on which no processing should occur.

Node Fails During Configuration with “869” LED Display

Problem

The system appears to be hung. “869” is displayed continuously on the system LED display.

Solution

A number of situations can cause this display to occur. Make sure all devices connected to the SCSI bus have unique SCSI IDs to avoid SCSI ID conflicts. In particular, check that the adapters and devices on each cluster node connected to the SCSI bus have a different SCSI ID. By default, AIX assigns an ID of 7 to a SCSI adapter when it configures the adapter. See the *Planning Guide* for more information on checking and setting SCSI IDs.

Node Cannot Rejoin Cluster after Being Dynamically Removed

Problem

A node that has been dynamically removed from a cluster cannot rejoin.

Solution

When you remove a node from the cluster, the cluster definition remains in the node’s Configuration Database. If you start cluster services on the removed node, the node reads this cluster configuration data and attempts to rejoin the cluster from which it had been removed. The other nodes no longer recognize this node as a member of the cluster and refuse to allow the node to join. Because the node requesting to join the cluster has the same cluster name as the existing cluster, it can cause the cluster to become unstable or crash the existing nodes.

To ensure that a removed node cannot be restarted with outdated Configuration Database information, complete the following procedure to remove the cluster definition from the node:

1. Stop cluster services on the node to be removed using the following command:

   ```
   clstop -R
   ```

   WARNING: You must stop cluster services on the node before removing it from the cluster.

   The **-R** flag removes the HACMP entry in the **/etc/inittab** file, preventing cluster services from being automatically started when the node is rebooted.

2. Remove the HACMP entry from the **rc.net** file using the following command:

```
clchipat false
```

3. Remove the cluster definition from the node's Configuration Database using the following command:

```
clrmclstr
```

You can also perform this task by selecting **Extended Configuration > Extended Topology Configuration > Configure an HACMP Cluster > Remove an HACMP Cluster** from the SMIT panel.

Resource Group Migration Is Not Persistent after Cluster Startup

Problem

You have specified a resource group migration operation using the Resource Group Migration Utility, in which you have requested that this particular migration **Persists across Cluster Reboot**, by setting this flag to **true** (or, by issuing the **clRGmove -p** command). Then, after you stopped and restarted the cluster services, this policy is *not* followed on one of the nodes in the cluster.

Solution

This problem occurs if, when you specified the persistent resource group migration, a node was down and inaccessible. In this case, the node did *not* obtain information about the persistent resource group migration, and, if after the cluster services are restarted, this node is the *first* to join the cluster, it will have no knowledge of the **Persist across Cluster Reboot** setting. Thus, the resource group migration will *not* be persistent. To restore the persistent migration setting, you must again specify it in SMIT under the **Extended Resource Configuration > HACMP Resource Group Configuration** SMIT menu.

SP Cluster Does Not Startup after Upgrade to HACMP 5.4.1

Problem

The ODM entry for group "hacmp" is removed on SP nodes. This problem manifests itself as the inability to start the cluster or **clcomd** errors.

Solution

To further improve security, the HACMP Configuration Database (ODM) has the following enhancements:

- **Ownership**. All HACMP ODM files are owned by user root and group hacmp. In addition, all HACMP binaries that are intended for use by non-root users are also owned by user root and group hacmp.
- **Permissions**. All HACMP ODM files, except for the **hacmpdisksubsystem** file with 600 permissions, are set with 640 permissions (readable by user root and group hacmp, writable by user root). All HACMP binaries that are intended for use by non-root users are installed with 2555 permissions (readable and executable by all users, with the **setgid** bit turned on so that the program runs as group hacmp).

During the installation, HACMP creates the group "hacmp" on all nodes if it does *not* already exist. By default, group hacmp has permission to read the HACMP ODMs, but does *not* have any other special authority. For security reasons, it is recommended *not* to expand the authority of group hacmp.

If you use programs that access the HACMP ODMs directly, you may need to rewrite them if they are intended to be run by non-root users:

- All access to the ODM data by non-root users should be handled via the provided HACMP utilities.
- In addition, if you are using the PSSP File Collections facility to maintain the consistency of **/etc/group**, the new group “hacmp” that is created at installation time on the individual cluster nodes may be lost when the next file synchronization occurs.

  There are two possible solutions to this problem. Take one of the following actions before installing HACMP 5.4.1:

  a. Turn off PSSP File Collections synchronization of **/etc/group**

  or

  b. Ensure that group “hacmp” is included in the master **/etc/group** file and ensure that the change is propagated to all cluster nodes.

Verification Problems When Nodes Have Different Fileset Levels

When clusters have nodes at different fileset levels (such as **cluster.es.server.diag**), the **clverify** program can hang or dump the core. Generally, HACMP nodes have the same fileset level, but you can be more likely to run into this situation while doing a node-by-node rolling PTF upgrade. These types of errors will prevent successful cluster startup.

When starting your cluster in this situation, ignore verification errors. You can do this by entering the following SMIT path: **smitty hacmp** > **System Management (C-SPOC)** > **Manage HACMP Services** > **Start Cluster Services**.

Within this panel, change **Ignore verification errors?** (default **false)** to **true**.

You can then start your cluster and avoid the problematic **clverify** program.

Note: Make sure your nodes are at equal fileset levels as soon as possible to avoid having to perform this procedure. Ignoring verification errors should be avoided.

Disk and File System Issues

The following potential disk and file system issues are described here:

AIX Volume Group Commands Cause System Error Reports

Problem

The **redefinevg**, **varyonvg**, **lqueryvg**, and **syncvg** commands fail and report errors against a shared volume group during system restart. These commands send messages to the console when automatically varying on a shared volume group. When configuring the volume groups for the shared disks, **autovaryon at boot** was *not* disabled. If a node that is up owns the shared drives, other nodes attempting to vary on the shared volume group will display various varyon error messages.

Solution

When configuring the shared volume group, set the **Activate volume group AUTOMATICALLY at system restart?** field to **no** on the SMIT **System Management (C-SPOC) > HACMP Logical Volume Management > Shared Volume Groups > Create a Shared Volume Group** panel. After importing the shared volume group on the other cluster nodes, use the following command to ensure that the volume group on each node is *not* set to **autovaryon at boot**:

```
chvg -an vgname
```

Verification Fails on Clusters with Disk Heartbeating Networks

Problem 1

With clusters that have disk heartbeating networks configured, when running verification the HACMP software indicates that verification failed “PVIDs do *not* match” error message.

Solution 1

Run verification with verbose logging to view messages that indicate where the error occurred (for example, the node, device, or command). The verification utility uses verbose logging to write to the **/var/hacmp/clverify/clverify.log** file.

If the hdisks have been renumbered, the disk heartbeating network may no longer be valid. Remove the disk heartbeating network and redefine it.

Ensure that the disk heartbeating networks are configured on enhanced concurrent volume groups. You can convert an existing volume group to enhanced concurrent mode. For information about converting a volume group, see the chapter Managing Shared LVM Components in a Concurrent Access Environment in the *Administration Guide*.

After correcting the problem, select the **Verify and Synchronize HACMP Configuration** option to synchronize the cluster resources configuration across all nodes. Then select the **Start Cluster Services** option from the **System Management (C-SPOC) > Manage HACMP Services** SMIT panel to start the Cluster Manager.

varyonvg Command Fails on a Volume Group

Problem 1

The HACMP software (the **/var/hacmp/log/hacmp.out** file) indicates that the **varyonvg** command failed when trying to vary on a volume group.

Solution 1

Ensure that the volume group is *not* set to **autovaryon** on any node and that the volume group (unless it is in concurrent access mode) is *not* already varied on by another node.

The **lsvg -o** command can be used to determine whether the shared volume group is active. Enter:

```
lsvg volume_group_name
```

on the node that has the volume group activated, and check the **AUTO ON** field to determine whether the volume group is automatically set to be on. If **AUTO ON** is set to **yes**, correct this by entering

```
chvg -an volume_group_name
```

Problem 2

The volume group information on disk differs from that in the Device Configuration Data Base.

Solution 2

Correct the Device Configuration Data Base on the nodes that have incorrect information:

1. Use the **smit exportvg** fastpath to export the volume group information. This step removes the volume group information from the Device Configuration Data Base.
2. Use the **smit importvg** fastpath to import the volume group. This step creates a new Device Configuration Data Base entry directly from the information on disk. After importing, be sure to change the volume group to *not* **autovaryon** at the next system boot.
3. Use the SMIT **Problem Determination Tools > Recover From HACMP Script Failure** panel to issue the **clruncmd** command to signal the Cluster Manager to resume cluster processing.

Problem 3

The HACMP software indicates that the **varyonvg** command failed because the volume group could *not* be found.

Solution 3

The volume group is *not* defined to the system. If the volume group has been newly created and exported, or if a **mksysb** system backup has been restored, you must import the volume group. Follow the steps described in Problem 2 to verify that the correct volume group name is being referenced.

Problem 4

The HACMP software indicates that the **varyonvg** command failed because the logical volume <name> is incomplete.

Solution 4

This indicates that the forced varyon attribute is configured for the volume group in SMIT, and that when attempting a forced varyon operation, HACMP did *not* find a single complete copy of the specified logical volume for this volume group.

Also, it is possible that you requested a forced varyon operation but did *not* specify the **super strict** allocation policy for the mirrored logical volumes. In this case, the success of the **varyon** command is *not* guaranteed. For more information on the forced varyon functionality, see the

chapter Planning Shared LVM Components in the *Planning Guide* and the Forcing a Varyon of Volume Groups section in the chapter on Configuring HACMP Resource Groups (Extended) in the *Administration Guide*.

cl_nfskill Command Fails

Problem

The **/var/hacmp/log/hacmp.out** file shows that the **cl_nfskill** command fails when attempting to perform a forced unmount of an NFS-mounted file system. NFS provides certain levels of locking a file system that resists forced unmounting by the **cl_nfskill** command.

Solution

Make a copy of the **/etc/locks** file in a separate directory before executing the **cl_nfskill** command. Then delete the original **/etc/locks** file and run the **cl_nfskill** command. After the command succeeds, re-create the **/etc/locks** file using the saved copy.

cl_scdiskreset Command Fails

Problem

The **cl_scdiskreset** command logs error messages to the **/var/hacmp/log/hacmp.out** file. To break the reserve held by one system on a SCSI device, the HACMP disk utilities issue the **cl_scdiskreset** command. The **cl_scdiskreset** command may fail if back-level hardware exists on the SCSI bus (adapters, cables or devices) or if a SCSI ID conflict exists on the bus.

Solution

See the appropriate sections in Chapter 2: Using Cluster Log Files to check the SCSI adapters, cables, and devices. Make sure that you have the latest adapters and cables. The SCSI IDs for each SCSI device *must* be different.

fsck Command Fails at Boot Time

Problem

At boot time, AIX runs the **fsck** command to check all the file systems listed in **/etc/filesystems** with the **check=true** attribute. If it cannot check a file system, AIX reports the following error:

```
Filesystem Helper: 0506-519 Device open failed
```

Solution

For file systems controlled by HACMP, this message typically does *not* indicate a problem. The file system check fails because the volume group defining the file system is *not* varied on. The boot procedure does *not* automatically vary on HACMP-controlled volume groups.

To prevent this message, make sure that all the file systems under HACMP control do *not* have the **check=true** attribute in their **/etc/filesystems** stanzas. To delete this attribute or change it to **check=false**, edit the **/etc/filesystems** file.

System Cannot Mount Specified File Systems

Problem

The **/etc/filesystems** file has *not* been updated to reflect changes to log names for a logical volume. If you change the name of a logical volume after the file systems have been created for that logical volume, the **/etc/filesystems** entry for the log does *not* get updated. Thus when

trying to mount the file systems, the HACMP software tries to get the required information about the logical volume name from the old log name. Because this information has *not* been updated, the file systems cannot be mounted.

Solution

Be sure to update the **/etc/filesystems** file after making changes to logical volume names.

Cluster Disk Replacement Process Fails

Problem 1

You cannot complete the disk replacement process due to a **node_down** event.

Solution 1

Once the node is back online, export the volume group, then import it again before starting HACMP on this node.

Problem 2

The disk replacement process failed while the **replacepv** command was running.

Solution 2

Delete the **/tmp/replacepv** directory, and attempt the replacement process again.

You can also try running the process on another disk.

Problem 3

The disk replacement process failed with a “no free disks” message while VPATH devices were available for replacement.

Solution 3

Be sure to convert the volume group from VPATH devices to hdisks, and attempt the replacement process again. When the disk is replaced, convert hdisks back to the VPATH devices. For instructions, see the Convert SDD VPATH Device Volume Group to an ESS hdisk Device Volume Group section in the chapter on Managing Shared LVM Components in the *Administration Guide*.

Automatic Error Notification Fails with Subsystem Device Driver

Problem

You set up automatic error notification for the 2105 IBM Enterprise Storage System (ESS), expecting it to log errors when there is a volume group loss. (The Subsystem Device Driver handles the loss.) However, the error notification fails and you get error messages in the **cspoc.log** and the **smit.log**.

Solution

If you set up automatic error notification for the 2105 IBM Enterprise Storage System (ESS), which uses the Subsystem Device Driver, all PVIDs must be on VPATHS, or the error notification fails. To avoid this failure, convert all hdisks to VPATH devices.

File System Change Not Recognized by Lazy Update

Problem

If you change the name of a file system, or remove a file system and then perform a lazy update, lazy update does *not* run the **imfs -lx** command before running the **imfs** command. This may lead to a failure during fallover or prevent a successful restart of the HACMP cluster services.

Solution

Use the C-SPOC utility to change or remove file systems. This ensures that **imfs -lx** runs before **imfs** and that the changes are updated on all nodes in the cluster.

Error Reporting provides detailed information about inconsistency in volume group state across the cluster. If this happens, take manual corrective action. If the file system changes are *not* updated on all nodes, update the nodes manually with this information.

Network and Switch Issues

The following potential network and switch issues are described here:

Unexpected Network Interface Failure in Switched Networks

Problem

Unexpected network interface failures can occur in HACMP configurations using switched networks if the networks and the switches are incorrectly defined/configured.

Solution

Take care to configure your switches and networks correctly. See the section on considerations for switched networks in the *Planning Guide* for more information.

Troubleshooting VLANs

Problem

Interface failures in Virtual LAN networks (from now on referred to as VLAN, Virtual Local Area Network)

Solution

To troubleshoot VLAN interfaces defined to HACMP and detect an interface failure, consider these interfaces as interfaces defined on single adapter networks.

For information on single adapter networks and the use of the **netmon.cf** file, see Missing Entries in the /etc/hosts for the netmon.cf File May Prevent RSCT from Monitoring Networks.

In particular, list the network interfaces that belong to a VLAN in the **ping_client_list** variable in the **/usr/es/sbin/cluster/etc/clinfo.rc** script and run **clinfo**. This way, whenever a cluster event occurs, **clinfo** monitors and detects a failure of the listed network interfaces. Due to the nature of Virtual Local Area Networks, other mechanisms to detect the failure of network interfaces are *not* effective.

Cluster Nodes Cannot Communicate

Problem

If your configuration has two or more nodes connected by a single network, you may experience a partitioned cluster. A partitioned cluster occurs when cluster nodes cannot communicate. In normal circumstances, a service network interface failure on a node causes the Cluster Manager to recognize and handle a **swap_adapter** event, where the service IP label/address is replaced with another IP label/address. However, if no other network interface is available, the node becomes isolated from the cluster. Although the Cluster Managers on other nodes are aware of the attempted **swap_adapter** event, they cannot communicate with the now isolated (partitioned) node because no communication path exists.

A partitioned cluster can cause GPFS to lose quorum. For more information, see the Appendix on GPFS Cluster Configuration, in the *Installation Guide*.

Solution

Make sure your network is configured for no single point of failure.

Distributed SMIT Causes Unpredictable Results

Problem

Using the AIX utility DSMIT on operations other than starting or stopping HACMP cluster services, can cause unpredictable results.

Solution

DSMIT manages the operation of networked IBM System p™ processors. It includes the logic necessary to control execution of AIX commands on all networked nodes. Since a conflict with HACMP functionality is possible, use DSMIT only to start and stop HACMP cluster services.

Token-Ring Network Thrashes

Problem

A Token-Ring network cannot reach steady state unless all stations are configured for the same ring speed. One symptom of the adapters being configured at different speeds is a clicking sound heard at the MAU (multi-station access unit).

Solution

Configure all adapters for either 4 or 16 Mbps.

System Crashes Reconnecting MAU Cables after a Network Failure

Problem

A global network failure occurs and crashes all nodes in a four-node cluster after reconnecting MAUs (multi-station access unit). More specifically, if the cables that connect multiple MAUs are disconnected and then reconnected, all cluster nodes begin to crash.

This result happens in a configuration where three nodes are attached to one MAU (MAU1) and a fourth node is attached to a second MAU (MAU2). Both MAUs (1 and 2) are connected together to complete a Token-Ring network. If MAU1 is disconnected from the network, all cluster nodes can continue to communicate; however, if MAU2 is disconnected, node isolation occurs.

Solution

To avoid causing the cluster to become unstable, do *not* disconnect cables connecting multiple MAUs in a Token-Ring configuration.

TMSCSI Will Not Properly Reintegrate when Reconnecting Bus

Problem

If the SCSI bus is broken while running as a target mode SCSI network, the network will *not* properly reintegrate when reconnecting the bus.

Solution

The HACMP software may need to be restarted on all nodes attached to that SCSI bus. When target mode SCSI is enabled and the **cfgmgr** command is run on a particular machine, it will go out on the bus and create a target mode initiator for every node that is on the SCSI network. In a four-node cluster, when all four nodes are using the same SCSI bus, each machine will have three initiator devices (one for each of the other nodes).

In this configuration, use a maximum of four target mode SCSI networks. You would therefore use networks between nodes A and B, B and C, C and D, and D and A.

Target mode SCSI devices are *not* always properly configured during the AIX boot process. Ensure that all the tmscsi initiator devices are available on all nodes before bringing up the cluster. To do this run **lsdev -Cc tmscsi**, which returns:

```
tmscsix  Available 00-12-00-40 SCSI I/O Controller Initiator Device
```

where *x* identifies the particular tmscsi device. If the status is *not* "Available," run the **cfgmgr** command and check again.

Recovering from PCI Hot Plug NIC Failure

Problem

If an unrecoverable error causes a PCI hot-replacement process to fail, the NIC may be left in an unconfigured state and the node may be left in maintenance mode. The PCI slot holding the NIC and/or the new NIC may be damaged at this point.

Solution

User intervention is required to get the node back in fully working order. For more information, refer to the AIX *Managing Hot Plug Connectors from System Management Guide: Operating System and Devices*.

Unusual Cluster Events Occur in Non-Switched Environments

Problem

Some network topologies may *not* support the use of simple switches. In these cases, you should expect that certain events may occur for no apparent reason. These events may be:

- Cluster unable to form, either all or some of the time
- **swap_adapter** pairs
- **swap_adapter**, immediately followed by a **join_standby**
- **fail_standby** and **join_standby** pairs.

These events occur when ARP packets are delayed or dropped. This is correct and expected HACMP behavior, as HACMP is designed to depend on core protocols strictly adhering to their related RFCs.

For a review of basic HACMP network requirements, see the *Planning Guide.*

Solution

The following implementations may reduce or circumvent these events:

- Increase the Failure Detection Rate (FDR) to exceed the ARP retransmit time of 15 seconds, where typical values have been calculated as follows:

  FDR = (2+ * 15 seconds) + >5 = 35+ seconds (usually 45-60 seconds)

  "2+" is a number greater than one in order to allow multiple ARP requests to be generated. This is required so that at least one ARP response will be generated and received before the FDR time expires and the network interface is temporarily marked down, then immediately marked back up.

  Keep in mind, however, that the "true" fallover is delayed for the value of the FDR.
- Increase the ARP queue depth.

  If you increase the queue, requests that are dropped or delayed will be masked until network congestion or network quiescence (inactivity) makes this problem evident.
- Use a dedicated switch, with all protocol optimizations turned off. Segregate it into a physical LAN segment and bridge it back into the enterprise network.
- Use permanent ARP entries (IP address to MAC address bindings) for all network interfaces. These values should be set at boot time, and since none of the ROM MAC addresses are used, replacing network interface cards will be invisible to HACMP.

Note: The above four items simply describe how some customers have customized their unique enterprise network topology to provide the classic protocol environment (strict adherence to RFCs) that HACMP requires. IBM cannot guarantee HACMP will work as expected in these approaches, since none addresses the root cause of the problem. *If your network topology requires consideration of any of these approaches please contact the IBM Consult Line for assistance.*

Cannot Communicate on ATM Classic IP Network

Problem

If you cannot communicate successfully to a cluster network interface of type atm (a cluster network interface configured over a Classic IP client, check the following:

Solution

1. Check the client configuration. Check that the 20-Byte ATM address of the Classic IP server that is specified in the client configuration is correct, and that the interface is configured as a Classic IP client (svc-c) and *not* as a Classic IP server (svc-s).
2. Check that the ATM TCP/IP layer is functional. Check that the UNI version settings that are configured for the underlying ATM device and for the switch port to which this device is connected are compatible. It is recommended *not* to use the value **auto_detect** for either side.

   If the connection between the ATM device# and the switch is not functional on the ATM protocol layer, this can also be due to a hardware failure (NIC, cable, or switch).

   Use the **arp** command to verify this:

```
[bass][/]>arp -t atm -a

SVC - at0 on device atm1 -
==========================
at0(10.50.111.6)
39.99.99.99.99.99.99.0.0.99.99.1.1.8.0.5a.99.98.fc.0
  IP Addr        VPI:VCI Handle ATM Address
  server_10_50_111(10.50.111.255)        0:888   15
39.99.99.99.99.99.99.0.0.99.99.1.1.88.88.88.88.a0.11.0

SVC - at1 on device atm0 -
==========================
at1(10.50.120.6)
39.99.99.99.99.99.99.0.0.99.99.1.1.8.0.5a.99.99.c1.1
  IP Addr        VPI:VCI Handle ATM Address
  ?(0.0.0.0)        N/A     N/A    15
39.99.99.99.99.99.99.0.0.99.99.1.1.88.88.88.88.a0.20.0

SVC - at3 on device atm2 -
==========================
at3(10.50.110.6)        8.0.5a.99.00.c1.0.0.0.0.0.0.0.0.0.0.0.0.0.0
  IP Addr        VPI:VCI Handle ATM Address
  ?(0.0.0.0)        0:608   16
39.99.99.99.99.99.99.0.0.99.99.1.1.88.88.88.88.a0.10.0
```

In the example above, the client at0 is operational. It has registered with its server, server_10_50_111.

The client at1 is not operational, since it could not resolve the address of its Classic IP server, which has the hardware address 39.99.99.99.99.99.99.0.0.99.99.1.1.88.88.88.88.a0.11.0. However, the ATM layer is functional, since the 20 byte ATM address that has been constructed for the client at1 is correct. The first 13 bytes is the switch address, 39.99.99.99.99.99.99.0.0.99.99.1.1.

For client at3, the connection between the underlying device atm2 and the ATM switch is not functional, as indicated by the failure to construct the 20 Byte ATM address of at3. The first 13 bytes do not correspond to the switch address, but contain the MAC address of the ATM device corresponding to atm2 instead.

Cannot Communicate on ATM LAN Emulation Network

Problem

You are having problems communicating with an ATM LANE client.

Solution

Check that the LANE client is registered correctly with its configured LAN Emulation server. A failure of a LANE client to connect with its LAN Emulation server can be due to the configuration of the LAN Emulation server functions on the switch. There are many possible reasons.

1. Correct client configuration: Check that the 20 Byte ATM address of the LAN Emulation server, the assignment to a particular ELAN, and the Maximum Frame Size value are all correct.
2. Correct ATM TCP/IP layer: Check that the UNI version settings that are configured for the underlying ATM device and for the switch port to which this device is connected are compatible. It is recommended not to use the value **auto_detect** for either side.

   If the connection between the ATM device# and the switch is not functional on the ATM protocol layer, this can also be due to a hardware failure (NIC, cable, or switch).

   Use the **enstat** and **tokstat** commands to determine the state of ATM LANE clients.

   ```
   bass][/]> entstat -d ent3
   ```

   The output will contain the following:

   ```
   General Statistics:
   -------------------
   No mbuf Errors: 0
   Adapter Reset Count: 3
   Driver Flags: Up Broadcast Running
           Simplex AlternateAddress

   ATM LAN Emulation Specific Statistics:
   --------------------------------------
   Emulated LAN Name: ETHER3
   Local ATM Device Name: atm1
   Local LAN MAC Address:
   42.0c.01.03.00.00
   Local ATM Address:
   39.99.99.99.99.99.99.00.00.99.99.01.01.08.00.5a.99.98.fc.04
   Auto Config With LECS:
   No
   LECS ATM Address:
   00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00
   LES ATM Address:
   39.99.99.99.99.99.99.00.00.99.99.01.01.88.88.88.88.00.03.00
   ```

   In the example above, the client is operational as indicated by the **Running flag**.

   If the client had failed to register with its configured LAN Emulation Server, the Running flag would not appear, instead the flag **Limbo** would be set.

   If the connection of the underlying device atm# was not functional on the ATM layer, then the local ATM address would not contain as the first 13 Bytes the Address of the ATM switch.

3. Switch-specific configuration limitations: Some ATM switches do not allow more than one client belonging to the same ELAN and configured over the same ATM device to register with the LAN Emulation Server at the same time. If this limitation holds and two clients are configured, the following are typical symptoms.
   - Cyclic occurrence of events indicating network interface failures, such as fail_standby, join_standby, and swap_adapter

     This is a typical symptom if two such clients are configured as cluster network interfaces. The client which first succeeds in registering with the LES will hold the connection for a specified, configuration-dependent duration. After it times out the other client succeeds in establishing a connection with the server, hence the cluster network interface configured on it will be detected as alive, and the former as down.
   - Sporadic events indicating an network interface failure (fail_standby, join_standby, and swap_adapter)

     If one client is configured as a cluster network interface and the other outside, this configuration error may go unnoticed if the client on which the cluster network interface is configured manages to register with the switch, and the other client remains inactive. The second client may succeed at registering with the server at a later moment, and a failure would be detected for the cluster network interface configured over the first client.

IP Label for HACMP Disconnected from AIX Interface

Problem

When you define network interfaces to the cluster configuration by entering or selecting an HACMP IP label, HACMP discovers the associated AIX network interface name. HACMP expects this relationship to remain unchanged. If you change the name of the AIX network interface name after configuring and synchronizing the cluster, HACMP will not function correctly.

Solution

If this problem occurs, you can reset the network interface name from the SMIT HACMP **System Management (C-SPOC)** panel. For more information, see the chapter on Managing the Cluster Resources in the *Administration Guide*.

TTY Baud Rate Setting Wrong

Problem

The default baud rate is 38400. Some modems or devices are incapable of doing 38400. If this is the case for your situation, you can change the default by customizing the RS232 network module to read the desired baud rate (9600/19200/38400).

Solution

Change the Custom Tuning Parameters for the RS232 network module. for instructions, see the chapter on Managing the Cluster Topology in the *Administration Guide*.

First Node Up Gives Network Error Message in hacmp.out

Problem

The first node up in a HACMP cluster gives the following network error message in **/var/hacmp/log/hacmp.out**, even if the network is healthy:

```
Error: EVENT START: network_down -1 ELLA
```

Whether the network is functional or not, the RSCT topology services heartbeat interval expires, resulting in the logging of the above error message. This message is only relevant to non-IP networks (such as RS232, TMSCSI, TMSSA). This behavior does not occur for disk heartbeating networks (for which **network_down** events are not logged in general).

Solution

Ignore the message and let the cluster services continue to function. You should see this error message corrected in a healthy cluster as functional network communication is eventually established between other nodes in the cluster. A **network_up** event will be run after the second node that has an interface on this network joins the cluster. If cluster communication is *not* established after this error message, then the problem should be diagnosed in other sections of this guide that discuss network issues.

Network Interface Card and Network ODMs Out of Sync with Each Other

Problem

In some situations, it is possible for the HACMPadapter or the HACMPnetwork ODMs to become out of sync with the AIX ODMs. For example, HACMP may refer to an Ethernet network interface card while AIX refers to a Token-Ring network interface card.

Note: This type of out-of-sync condition only occurs as a result of the following situations:

- If the hardware settings have been adjusted *after* the HACMP cluster has been successfully configured and synchronized

or

- If the wrong values were selected when configuring predefined communication interfaces to HACMP.

Solution

Run cluster verification to detect and report the following network and network interface card type incompatibilities:

- The network interface card configured in HACMP is the correct one for the node's hardware
- The network interface cards configured in HACMP and AIX match each other.

If verification returns an error, examine and adjust the selections made on the **Extended Configuration > Extended Topology Configuration > Configuring HACMP Communication Interfaces/Devices > Change/Show Communication Interfaces/Devices** SMIT panel. For more information on this screen, see Chapter 4: Configuring HACMP Cluster Topology and Resources (Extended) of the *Administration Guide*.

Non-IP Network, Network Adapter or Node Failures

Problem

The non-IP interface declares its neighbor down after the Failure Detection Rate has expired for that network interface type. HACMP waits the same interval again before declaring the local interface down (if no heartbeat is received from the neighbor).

Solution

The non-IP heartbeating helps determine the difference between a NIC failure, network failure, and even more importantly node failure. When a non-IP network failure occurs, HACMP detects a non-IP network down and logs an error message in the **/var/hacmp/log/hacmp.out** file.

Use the **clstat -s** command to display the service IP labels for non-IP networks that are currently down on a network.

The RSCT **topsvcs** daemon logs messages whenever an interface changes state. These errors are visible in the **errpt**.

For more information, see the section Changing the Configuration of a Network Module in the chapter on Managing the Cluster Topology in the *Administration Guide*.

Networking Problems Following HACMP Fallover

Problem

If you are using Hardware Address Takeover (HWAT) with any gigabit Ethernet adapters supporting flow control, you may be exposed to networking problems following an HACMP fallover. If a system crash occurs on one node and power still exists for the adapters on the crashed node, even though the takeover is successful, the network connection to the takeover node may be lost or the network containing the failing adapters may lock up. The problem is related to flow control being active on the gigabit Ethernet adapter in conjunction with how the Ethernet switch handles this situation.

Solution

Turn off flow control on the gigabit Ethernet adapters.

To disable flow control on the adapter type:

```
ifconfig entX detach
    # where entX corresponds to the gigabit adapter device
chdev -l entX -a flow_ctrl=no
```

Then reconfigure the network on that adapter.

Packets Lost during Data Transmission

Problem

If data is intermittently lost during transmission, it is possible that the maximum transmission unit (MTU) has been set to different sizes on different nodes. For example, if Node A sends 8 K packets to Node B, which can accept 1.5 K packets, Node B assumes the message is complete; however data may have been lost.

Solution

Run the cluster verification utility to ensure that all of the network interface cards on all cluster nodes during the same network have the same setting for MTU size. If the MTU size is inconsistent across the network, an error displays, which enables you to determine which nodes to adjust.

Note: You can change an MTU size by using the following command:

```
chev -l en0 -a mtu=<new_value_from_1_to_8>
```

Verification Fails when Geo Networks Uninstalled

Problem

HAGEO uninstalled, but Geo network definitions remain and cluster verification fails.

Solution

After HAGEO is uninstalled, any HACMP networks which are still defined as type Geo_Primary or Geo_Secondary must either be removed, or their type must be modified to correspond to the network type (such as Ethernet, Token Ring, RS232). HACMP verification will fail unless these changes are made to the HACMP network definitions.

Missing Entries in the /etc/hosts for the netmon.cf File May Prevent RSCT from Monitoring Networks

Problem

Missing entries in the **/etc/hosts** for the **netmon.cf** file may prevent your networks from being properly monitored by the **netmon** utility of the RSCT Topology Services.

Solution

Make sure to include the entries for the **netmon.cf** file – each IP address and its corresponding label – in the **/etc/hosts** file. If the entries are missing, it may result in the NIM process of RSCT being blocked while RSCT attempts to determine the state of the local adapters.

In general, we recommend to create the **netmon.cf** file for the cluster configurations where there are networks that under certain conditions can become single adapter networks. In such networks, it can be difficult for HACMP to accurately determine adapter failure. This is because RSCT Topology Services cannot force packet traffic over the single adapter to verify its operation. The creation of the **netmon.cf** file allows RSCT to accurately determine adapter failure.

For more information on creating the **netmon.cf** file, see the *Planning Guide*.

Cluster Communications Issues

The following potential cluster communications issues are described here:

- Message Encryption Fails
- Cluster Nodes Do Not Communicate with Each Other.

Message Encryption Fails

Problem

If you have message authentication or message authentication and encryption enabled, and you receive a message that encryption fails or that a message could not be decrypted.

Solution

If the encryption filesets are not found on the local node, a message indicates that the encryption libraries were not found.

If you did not receive a message that encryption libraries could not be found on the local node, check the **clcomd.log** file to determine if the encryption filesets are not found on a remote node.

Verify whether the cluster node has the following filesets installed:

- For data encryption with DES message authentication: **rsct.crypt.des**
- For data encryption standard Triple DES message authentication: **rsct.crypt.3des**
- For data encryption with Advanced Encryption Standard (AES) message authentication: **rsct.crypt.aes256**.

If needed, install these filesets from the AIX Expansion Pack CD-ROM.

If the filesets are installed after HACMP is already running, start and stop the HACMP Cluster Communications daemon to enable HACMP to use these filesets. To restart the Cluster Communications daemon:

```
stopscr -s clcomdes
startsrc -s clcomdes
```

If the filesets are present, and you get an encryption error, the encryption filesets may have been installed, or reinstalled, after HACMP was running. In this case, restart the Cluster Communications daemon as described above.

Cluster Nodes Do Not Communicate with Each Other

Problem

Cluster nodes are unable to communicate with each, and you have one of the following configured:

- Message authentication, or message authentication and encryption enabled
- Use of persistent IP labels for VPN tunnels.

Solution

Make sure that the network is operational, see the section Network and Switch Issues.

Check if the cluster has persistent IP labels. If it does, make sure that they are configured correctly and that you can ping the IP label.

If you are using message authentication, or message authentication and encryption:

- Make sure that each cluster node has the same setting for message authentication mode. If the modes are different, on each node set message authentication mode to None and configure message authentication again.
- Make sure that each node has the same type of encryption key in the **/usr/es/sbin/cluster/etc** directory. Encryption keys cannot reside in other directories.

If you have configured use of persistent IP labels for a VPN:

1. Change **User Persistent Labels** to **No**.
2. Synchronize cluster configuration.
3. Change **User Persistent Labels** to **Yes**.

HACMP Takeover Issues

Note that if you are investigating resource group movement in HACMP—for instance, investigating why an **rg_move** event has occurred—always check the **/var/hacmp/log/hacmp.out** file. In general, given the recent changes in the way resource groups are handled and prioritized in fallover circumstances, particularly in HACMP, the **hacmp.out** file and its event summaries have become even more important in tracking the activity and resulting location of your resource groups. In addition, with parallel processing of resource groups, the **hacmp.out** file reports details that cannot be seen in the cluster history log or the **clstrmgr.debug** log file. Always check the **hacmp.out** log early on when investigating resource group movement after takeover activity.

The following potential takeover issues are described here:

varyonvg Command Fails During Takeover

Problem

The HACMP software failed to vary on a shared volume group. The volume group name is either missing or is incorrect in the HACMP Configuration Database object class.

Solution

- Check the **/var/hacmp/log/hacmp.out** file to find the error associated with the varyonvg failure.
- List all the volume groups known to the system using the **lsvg** command; then check that the volume group names used in the HACMPresource Configuration Database object class are correct. To change a volume group name in the Configuration Database, from the main HACMP SMIT panel select **Initialization and Standard Configuration > Configure HACMP Resource Groups > Change/Show Resource Groups**, and select the resource group where you want the volume group to be included. Use the **Volume Groups** or **Concurrent Volume Groups** fields on the **Change/Show Resources and Attributes for a Resource Group** panel to set the volume group names. After you correct the problem, use the SMIT **Problem Determination Tools > Recover From HACMP Script Failure** panel to issue the **clruncmd** command to signal the Cluster Manager to resume cluster processing.
- Run the cluster verification utility to verify cluster resources.

Highly Available Applications Fail

Problem 1

Highly available applications fail to start on a fallover node after an IP address takeover. The hostname may not be set.

Solution 1

Some software applications require an exact hostname match before they start. If your HACMP environment uses IP address takeover and starts any of these applications, add the following lines to the script you use to start the application servers:

```
mkdev -t inet
chdev -l inet0 -a hostname=nnn
```

where *nnn* is the hostname of the machine the fallover node is masquerading as.

Problem 2

An application that a user has manually stopped following a stop of cluster services where resource groups were placed in an UNMANAGED state, does not restart with reintegration of the node.

Solution 2

Check that the relevant application entry in the /**usr/es/sbin/cluster/server.status** file has been removed prior to node reintegration.

Since an application entry in the /**usr/es/sbin/cluster/server.status** file lists all applications already running on the node, HACMP will not restart the applications with entries in the **server.status** file.

Deleting the relevant application **server.status** entry before reintegration, allows HACMP to recognize that the highly available application is not running, and that it must be restarted on the node.

Node Failure Detection Takes Too Long

Problem

The Cluster Manager fails to recognize a node failure in a cluster configured with a Token-Ring network. The Token-Ring network cannot become stable after a node failure unless the Cluster Manager allows extra time for failure detection.

In general, a buffer time of 14 seconds is used before determining failures on a Token-Ring network. This means that all Cluster Manager failure modes will take an extra 14 seconds if the Cluster Manager is dealing with Token-Ring networks. This time, however, does not matter if the Cluster Manager is using both Token-Ring and Ethernet. If Cluster Manager traffic is using a Token-Ring network interface, the 14 extra seconds for failures applies.

Solution

If the extra time is not acceptable, you can switch to an alternative network, such as an Ethernet. Using a non-IP heartbeating network (such as RS232) as recommended for all clusters should prevent this problem.

For some configurations, it is possible to run all the cluster network traffic on a separate network (Ethernet), even though a Token-Ring network also exists in the cluster. When you configure the cluster, include only the interfaces used on this separate network. Do not include the Token-Ring interfaces.

Since the Cluster Manager has no knowledge of the Token-Ring network, the 14-second buffer does not apply; thus failure detection occurs faster. Since the Cluster Manager does not know about the Token-Ring network interfaces, it cannot monitor them, nor can it swap network interfaces if one of the network interfaces fails or if the cables are unplugged.

HACMP Selective Fallover Is Not Triggered by a Volume Group Loss of Quorum Error in AIX

Problem

HACMP fails to selectively move the affected resource group to another cluster node when a volume group quorum loss occurs.

Solution

If quorum is lost for a volume group that belongs to a resource group on a cluster node, the system checks whether the LVM_SA_QUORCLOSE error appeared in the node's AIX error log file and informs the Cluster Manager to selectively move the affected resource group. HACMP uses this error notification method only for mirrored volume groups with quorum enabled.

If fallover does not occur, check that the LVM_SA_QUORCLOSE error appeared in the AIX error log. When the AIX error log buffer is full, new entries are discarded until buffer space becomes available and an error log entry informs you of this problem. To resolve this issue, increase the size of the AIX error log internal buffer for the device driver. For information about increasing the size of the error log buffer, see the AIX documentation listed in About This Guide.

Group Services Sends GS_DOM_MERGE_ER Message

Problem

A Group Services merge message is displayed and the node receiving the message shuts itself down. You see a GS_DOM_MERGE_ER error log entry, as well as a message in the Group Services daemon log file:

```
"A better domain XXX has been discovered, or domain master
requested to dissolve the domain."
```

A Group Services merge message is sent when a node loses communication with the cluster and then tries to reestablish communication.

Solution

Because it may be difficult to determine the state of the missing node and its resources (and to avoid a possible data divergence if the node rejoins the cluster), you should shut down the node and successfully complete the takeover of its resources.

For example, if a cluster node becomes unable to communicate with other nodes, yet it continues to work through its process table, the other nodes conclude that the "missing" node has failed because they no longer are receiving keepalive messages from the "missing" node. The remaining nodes then process the necessary events to acquire the disks, IP addresses, and other resources from the "missing" node. This attempt to take over resources results in the dual-attached disks receiving resets to release them from the "missing" node and to start IP address takeover scripts.

As the disks are being acquired by the takeover node (or after the disks have been acquired and applications are running), the "missing" node completes its process table (or clears an application problem) and attempts to resend keepalive messages and rejoin the cluster. Since the disks and IP address have been successfully taken over, it becomes possible to have a duplicate IP address on the network and the disks may start to experience extraneous traffic on the data bus.

Because the reason for the "missing" node remains undetermined, you can assume that the problem may repeat itself later, causing additional downtime of not only the node but also the cluster and its applications. Thus, to ensure the highest cluster availability, GS merge messages should be sent to any "missing" cluster node to identify node isolation, to permit the successful takeover of resources, and to eliminate the possibility of data corruption that can occur if both the takeover node and the rejoining "missing" node attempt to write to the disks. Also, if two nodes exist on the network with the same IP address, transactions may be missed and applications may hang.

When you have a partitioned cluster, the node(s) on each side of the partition detect this and run a **node_down** for the node(s) on the opposite side of the partition. If while running this or after communication is restored, the two sides of the partition do not agree on which nodes are still members of the cluster, a decision is made as to which partition should remain up, and the other partition is shutdown by a GA merge from nodes in the other partition or by a node sending a GS merge to itself.

In clusters consisting of more than two nodes the decision is based on which partition has the most nodes left in it, and that partition stays up. With an equal number of nodes in each partition (as is always the case in a two-node cluster) the node(s) that remain(s) up is determined by the node number (lowest node number in cluster remains) which is also generally the first in alphabetical order.

Group Services domain merge messages indicate that a node isolation problem was handled to keep the resources as highly available as possible, giving you time to later investigate the problem and its cause. When a domain merge occurs, Group Services and the Cluster Manager exit. The **clstrmgr.debug** file will contain the following error:

```
"announcementCb: GRPSVCS announcement code=n; exiting"
"CHECK FOR FAILURE OF RSCT SUBSYSTEMS (topsvcs or grpsvcs)"
```

cfgmgr Command Causes Unwanted Behavior in Cluster

Problem

SMIT commands like **Configure Devices Added After IPL** use the **cfgmgr** command. Sometimes this command can cause unwanted behavior in a cluster. For instance, if there has been a network interface swap, the **cfgmgr** command tries to reswap the network interfaces, causing the Cluster Manager to fail.

Solution

See the *Installation Guide* for information about modifying **rc.net**, thereby bypassing the issue. You can use this technique at all times, not just for IP address takeover, but it adds to the overall takeover time, so it is not recommended.

Releasing Large Amounts of TCP Traffic Causes DMS Timeout

Large amounts of TCP traffic over an HACMP-controlled service interface may cause AIX to experience problems when queuing and later releasing this traffic. When traffic is released, it generates a large CPU load on the system and prevents timing-critical threads from running, thus causing the Cluster Manager to issue a deadman switch (DMS) timeout.

To reduce performance problems caused by releasing large amounts of TCP traffic into a cluster environment, consider increasing the Failure Detection Rate beyond **Slow** to a time that can handle the additional delay before a takeover. See the Changing the Failure Detection Rate of a Network Module section in the chapter on Managing the Cluster Topology in the *Administration Guide*.

Also, to lessen the probability of a DMS timeout, complete the following steps before issuing a **node_down**:

1. Use the **netstat** command to identify the ports using an HACMP-controlled service network interface.
2. Use the **ps** command to identify all remote processes logged to those ports.
3. Use the **kill** command to terminate these processes.

Deadman Switch Causes a Node Failure

Problem

The node experienced an extreme performance problem, such as a large I/O transfer, excessive error logging, or running out of memory, and the Topology Services daemon (**hatsd**) is starved for CPU time. It could not reset the deadman switch within the time allotted. Misbehaved applications running at a priority higher than the Cluster Manager can also cause this problem.

Solutions

The deadman switch describes the AIX kernel extension that causes a system panic and dump under certain cluster conditions if it is not reset. The deadman switch halts a node when it enters a hung state that extends beyond a certain time limit. This enables another node in the cluster to acquire the hung node's resources in an orderly fashion, avoiding possible contention problems. Solutions related to performance problems should be performed in the following order:

1. Tune the system using I/O pacing and increasing the **syncd** frequency as directed in the chapter on Configuring AIX for HACMP in the *Installation Guide.*
2. If needed, increase the amount of memory available for the communications subsystem.
3. Tune virtual memory management (VMM). This is explained below.
4. Change the Failure Detection Rate. For more information, see the Changing the Failure Detection Rate of a Network Module section in the chapter on Managing the Cluster Topology in the *Administration Guide*.

Tuning Virtual Memory Management

For most customers, increasing *minfree/maxfree* whenever the freelist gets below *minfree* by more than 10 times the number of memory pools is necessary to allow a system to maintain consistent response times. To determine the current size of the freelist, use the **vmstat** command. The size of the freelist is the value labeled *free*. The number of memory pools in a system is the maximum of the number of CPUs/8 or memory size in GB/16, but never more than the number of CPUs and always at least one. The value of *minfree* is shown by the **vmtune** command.

In systems with multiple memory pools, it may also be important to increase *minfree/maxfree* even though *minfree* will not show as 120, since the default *minfree* is 120 times the number of memory pools. If raising *minfree/maxfree* is going to be done, it should be done with care, that is, not setting it too high since this may mean too many pages on the freelist for no real reason. One suggestion is to increase *minfree* and *maxfree* by 10 times the number of memory pools, then observe the freelist again. In specific application environments, such as multiple processes (three or more) each reading or writing a very large sequential file (at least 1GB in size each) it may be best to set *minfree* relatively high, e.g. 120 times the number of CPUs, so that maximum throughput can be achieved.

This suggestion is specific to a multi-process large sequential access environment. *Maxfree*, in such high sequential I/O environments, should also be set more than just 8 times the number of CPUs higher than *minfree*, e.g. *maxfree* = *minfree* + (*maxpgahead* x the number of CPUs), where *minfree* has already been determined using the above formula. The default for *maxpgahead* is 8, but in many high sequential activity environments, best performance is achieved with *maxpgahead* set to 32 or 64. This suggestion applies to all System p™ models

still being marketed, regardless of memory size. Without these changes, the chances of a DMS timeout can be high in these specific environments, especially those with minimum memory size.

For database environments, these suggestions should be modified. If JFS files are being used for database tables, then watching *minfree* still applies, but *maxfree* could be just *minfree* + (8 x the number of memory pools). If raw logical volumes are being used, the concerns about *minfree/maxfree* do not apply, but the following suggestion about *maxperm* is relevant.

In any environment (HA or otherwise) that is seeing non-zero paging rates, it is recommended that *maxperm* be set lower than the default of ~80%. Use the *avm* column of **vmstat** as an estimate of the number of working storage pages in use, or the number of valid memory pages, (should be observed at full load on the system's real memory, as shown by **vmtune**) to determine the percentage of real memory occupied by working storage pages. For example, if *avm* shows as 70% of real memory size, then *maxperm* should be set to 25% (**vmtune** -P 25). The basic formula used here is *maxperm* = 95 - *avm*/memory size in pages. If *avm* is less than or equal to 95% of memory, then this system is memory constrained. The options at this point are to set *maxperm* to 5% and incur some paging activity, add additional memory to this system, or to reduce the total workload run simultaneously on the system so that *avm* is lowered.

Deadman Switch Time to Trigger

The Topology Services chapter in the *Parallel System Support Programs for AIX Diagnosis Guide* has several hints about how to avoid having the **hatsd** blocked which causes the deadman switch (DMS) to hit. The relevant information is in the Diagnostic Procedure section of the chapter. See "Action 5 - Investigate hatsd problem" and "Action 8 - Investigate node crash". The URL for this Guide follows:

http://publibfp.boulder.ibm.com/epubs/pdf/a2273503.pdf

Running the /usr/sbin/rsct/bin/hatsdmsinfo command

This command is useful for checking on the deadman switch trigger time.

Output of the **/usr/sbin/rsct/bin/hatsdmsinfo** command looks like this:

```
============================================================
Information for Topology Services -- HACMP /ES
DMS Trigger time: 20.000 seconds.
Last DMS Resets                          Time to Trigger (seconds)
06/04/02 06:51:53.064                    19.500
06/04/02 06:51:53.565                    19.499
06/04/02 06:51:54.065                    19.500
06/04/02 06:51:54.565                    19.500
06/04/02 06:51:55.066                    19.500
06/04/02 06:51:55.566                    19.499
DMS Resets with small time-to-trigger    Time to Trigger (seconds)
Threshold value: 15.000 seconds.
```

A "device busy" Message Appears after node_up_local Fails

Problem

A *device busy* message in the **/var/hacmp/log/hacmp.out** file appears when swapping hardware addresses between the boot and service address. Another process is keeping the device open.

Solution

Check to see if **sysinfod,** the SMUX peer daemon, or another process is keeping the device open. If it is **sysinfod**, restart it using the **-H** option.

Network Interfaces Swap Fails Due to an rmdev “device busy” Error

Problem

Network interfaces swap fails due to an **rmdev** *device busy* error. For example, **/var/hacmp/log/hacmp.out** shows a message similar to the following:

```
Method error (/etc/methods/ucfgdevice):
0514-062 Cannot perform the requested function because the specified
device is busy.
```

Solution

Check to see whether the following applications are being run on the system. These applications may keep the device busy:

- **SNA**

  Use the following commands to see if SNA is running:

  ```
  lssrc -g sna
  ```

  Use the following command to stop SNA:

  ```
  stopsrc -g sna
  ```

  If that does not work, use the following command:

  ```
  stopsrc -f -s sna
  ```

  If that does not work, use the following command:

  ```
  /usr/bin/sna -stop sna -t forced
  ```

  If that does not work, use the following command:

  ```
  /usr/bin/sna -stop sna -t cancel
  ```

- **Netview / Netmon**

  Ensure that the **sysmond** daemon has been started with a **-H** flag. This will result in opening and closing the network interface each time SM/6000 goes out to read the status, and allows the **cl_swap_HW_address** script to be successful when executing the **rmdev** command after the **ifconfig detach** before swapping the hardware address.

  Use the following command to stop all Netview daemons:

  ```
  /usr/OV/bin/nv6000_smit stopdaemons
  ```

- **IPX**

  Use the following commands to see if IPX is running:

  ```
  ps -ef |grep npsd
  ps -ef |grep sapd
  ```

  Use the following command to stop IPX:

  ```
  /usr/lpp/netware/bin/stopnps
  ```

- **NetBIOS.**

  Use the following commands to see if NetBIOS is running:

  ```
  ps -ef | grep netbios
  ```

Use the following commands to stop NetBIOS and unload NetBIOS streams:

```
mcsadm stop; mcs0 unload
```

- Unload various streams if applicable (that is, if the file exists):

```
cd /etc
strload -uf /etc/dlpi.conf
strload -uf /etc/pse.conf
strload -uf /etc/netware.conf
strload -uf /etc/xtiso.conf
```

- Some customer applications will keep a device busy. Ensure that the shared applications have been stopped properly.

MAC Address Is Not Communicated to the Ethernet Switch

Problem

With switched Ethernet networks, MAC address takeover sometimes appears to not function correctly. Even though HACMP has changed the MAC address of the network interface, the switch is not informed of the new MAC address. The switch does not then route the appropriate packets to the network interface.

Solution

Do the following to ensure that the new MAC address is communicated to the switch:

1. Modify the line in **/usr/es/sbin/cluster/etc/clinfo.rc** that currently reads:

   ```
   PING_CLIENT_LIST=" "
   ```

2. Include on this line the names or IP addresses of at least one client on each subnet on the switched Ethernet.
3. Run **clinfoES** on all nodes in the HACMP cluster that are attached to the switched Ethernet.

   If you normally start HACMP cluster services using the **/usr/es/sbin/cluster/etc/rc.cluster** shell script, specify the **-i** option. If you normally start HACMP cluster services through SMIT, specify **yes** in the **Start Cluster Information Daemon?** field.

Client Issues

The following potential HACMP client issues are described here:

- Network Interface Swap Causes Client Connectivity Problem
- Clients Cannot Access Applications
- Clients Cannot Find Clusters
- Clinfo Does Not Appear to Be Running
- Clinfo Does Not Report That a Node Is Down.

Network Interface Swap Causes Client Connectivity Problem

Problem

The client cannot connect to the cluster. The ARP cache on the client node still contains the address of the failed node, not the fallover node.

Solution

Issue a **ping** command to the client from a cluster node to update the client's ARP cache. Be sure to include the client name as the argument to this command. The **ping** command will update a client's ARP cache even if the client is not running **clinfoES**. You may need to add a call to the **ping** command in your application's pre- or post-event processing scripts to automate this update on specific clients. Also consider using hardware address swapping, since it will maintain configured hardware-to-IP address mapping within your cluster.

Clients Cannot Access Applications

Problem

The **SNMP process** failed.

Solution

Check the **/etc/hosts** file on the node on which **SNMP-**failed to ensure that it contains IP labels or addresses of cluster nodes. Also see Clients Cannot Find Clusters.

Clients Cannot Find Clusters

Problem

The **clstat** utility running on a client cannot find any clusters. The **clinfoES** daemon has not properly managed the data structures it created for its clients (like **clstat**) because it has not located an SNMP process with which it can communicate. Because **clinfoES** obtains its cluster status information from SNMP, it cannot populate the HACMP MIB if it cannot communicate with this daemon. As a result, a variety of intermittent problems can occur between SNMP and **clinfoES**.

Solution

Create an updated client-based **clhosts** file by running verification with automatic corrective actions enabled. This produces a **clhosts.client** file on the server nodes. Copy this file to the **/usr/es/sbin/cluster/etc/** directory on the clients, renaming the file **clhosts**. The **clinfoES** daemon uses the addresses in this file to attempt communication with an SNMP process executing on an HACMP server.

WARNING: For non-alias IP networks do not include standby addresses in the **clhosts** file.

Also, check the **/etc/hosts** file on the node on which the SNMP process is running and on the node having problems with **clstat** or other **clinfo** API programs.

Clinfo Does Not Appear to Be Running

Problem

The service and boot addresses of the cluster node from which **clinfoES** was started do not exist in the client-based **clhosts** file.

Solution

Create an updated client-based **clhosts** file by running verification with automatic corrective actions enabled. This produces a **clhosts.client** file on the server nodes. Copy this file to the **/usr/es/sbin/cluster/etc/** directory on the clients, renaming the file **clhosts**. Then run the **clstat** command.

Clinfo Does Not Report That a Node Is Down

Problem

Even though the node is down, the SNMP daemon and **clinfoES** report that the node is up. All the node's interfaces are listed as down.

Solution

When one or more nodes are active and another node tries to join the cluster, the current cluster nodes send information to the SNMP daemon that the joining node is up. If for some reason, the node fails to join the cluster, **clinfoES** does not send another message to the SNMP daemon the report that the node is down.

To correct the cluster status information, restart the SNMP daemon, using the options on the HACMP Cluster Services SMIT panel.

Miscellaneous Issues

The following non-categorized HACMP issues are described here:

- Limited Output when Running the tail -f Command on /var/hacmp/log/hacmp.out
- CDE Hangs after IPAT on HACMP Startup
- Cluster Verification Gives Unnecessary Message
- config_too_long Message Appears
- Console Displays SNMP Messages
- Device LEDs Flash "888" (System Panic)
- Unplanned System Reboots Cause Fallover Attempt to Fail
- Deleted or Extraneous Objects Appear in NetView Map
- F1 Does not Display Help in SMIT Panels
- /usr/es/sbin/cluster/cl_event_summary.txt File (Event Summaries Display) Grows Too Large
- View Event Summaries Does Not Display Resource Group Information as Expected
- Application Monitor Problems
- Cluster Disk Replacement Process Fails
- Resource Group Unexpectedly Processed Serially
- rg_move Event Processes Several Resource Groups at Once
- File System Fails to Unmount
- Dynamic Reconfiguration Sets a Lock
- WebSMIT Does Not "See" the Cluster
- Problems with WPAR-Enabled Resource Group

Note that if you are investigating resource group movement in HACMP—for instance, investigating why an **rg_move** event has occurred—always check the **/var/hacmp/log/hacmp.out** file. In general, given the recent changes in the way resource groups are handled and prioritized in fallover circumstances, particularly in HACMP, the **hacmp.out** file and its event summaries have become even more important in tracking the activity and resulting location of your resource groups. In addition, with parallel processing of resource groups, the **hacmp.out** file reports details that will not be seen in the cluster history log or the **clstrmgr.debug** file. Always check this log early on when investigating resource group movement after takeover activity.

Limited Output when Running the tail -f Command on /var/hacmp/log/hacmp.out

Problem

Only script start messages appear in the **/var/hacmp/log/hacmp.out** file. The script specified in the message is not executable, or the DEBUG level is set to **low**.

Solution

Add executable permission to the script using the **chmod** command, and make sure the DEBUG level is set to **high**.

CDE Hangs after IPAT on HACMP Startup

Problem

If CDE is started before HACMP is started, it binds to the boot address. When HACMP is started it swaps the IP address to the service address. If CDE has already been started this change in the IP address causes it to hang.

Solution

- The output of ***hostname*** and the **uname -n** must be the same. If the output is different, use **uname -S *hostname*** to make the **uname** match the output from ***hostname***.
- Define an alias for the ***hostname*** on the loopback address. This can be done by editing **/etc/hosts** to include an entry for:

  ```
  127.0.0.1     loopback localhost hostname
  ```

  where ***hostname*** is the name of your host. If name serving is being used on the system edit the **/etc/netsvc.conf** file such that the local file is checked first when resolving names.
- Ensure that the ***hostname*** and the service IP label resolve to different addresses. This can be determine by viewing the output of the **/bin/host** command for both the ***hostname*** and the service IP label.

Cluster Verification Gives Unnecessary Message

Problem

You get the following message regardless of whether or not you have configured Auto Error Notification:

```
"Remember to redo automatic error notification if configuration
has changed."
```

Solution

Ignore this message if you have not configured Auto Error Notification.

config_too_long Message Appears

This message appears each time a cluster event takes more time to complete than a specified time-out period.

In versions prior to 4.5, the time-out period was fixed for all cluster events and set to 360 seconds by default. If a cluster event, such as a **node_up** or a **node_down** event, lasted longer than 360 seconds, then every 30 seconds HACMP issued a **config_too_long** warning message that was logged in the **hacmp.out** file.

In HACMP 4.5 and up, you can customize the time period allowed for a cluster event to complete before HACMP issues a system warning for it.

If this message appears, in the **hacmp.out** Event Start you see:

```
config_too_long $sec $event_name $argument
```

- `$event_name` is the reconfig event that failed
- `$argument` is the parameter(s) used by the event
- $sec is the number of seconds before the message was sent out.

In versions prior to HACMP 4.5, **config_too_long** messages continued to be appended to the **hacmp.out** file every 30 seconds until action was taken.

Starting with version 4.5, for each cluster event that does not complete within the specified event duration time, **config_too_long** messages are logged in the **hacmp.out** file and sent to the console according to the following pattern:

- The first five **config_too_long** messages appear in the **hacmp.out** file at 30-second intervals
- The next set of five messages appears at interval that is double the previous interval until the interval reaches one hour
- These messages are logged every hour until the event is complete or is terminated on that node.

This message could appear in response to the following problems:

Problem

Activities that the script is performing take longer than the specified time to complete; for example, this could happen with events involving many disks or complex scripts.

Solution

- Determine what is taking so long to execute, and correct or streamline that process if possible.
- Increase the time to wait before calling **config_too_long**.

  You can customize **Event Duration Time** using the **Change/Show Time Until Warning** panel in SMIT. Access this panel through the **Extended Configuration > Extended Event Configuration** SMIT panel.

For complete information on tuning event duration time, see the Tuning Event Duration Time Until Warning section in the chapter on Configuring Cluster Events in the *Administration Guide*.

Problem

A command is hung and event script is waiting before resuming execution. If so, you can probably see the command in the AIX process table **(ps -ef**). It is most likely the last command in the **/var/hacmp/log/hacmp.out** file, before the **config_too_long** script output.

Solution

You may need to kill the hung command. See also Dynamic Reconfiguration Sets a Lock.

Console Displays SNMP Messages

Problem

The **/etc/syslogd** file has been changed to send the **daemon.notice** output to **/dev/console**.

Solution

Edit the **/etc/syslogd** file to redirect the **daemon.notice** output to **/usr/tmp/snmpd.log**. The **snmpd.log** file is the default location for logging messages.

Device LEDs Flash “888” (System Panic)

Problem

The **crash** ***system dump device*** with **stat** subcommand indicates the panic was caused by the deadman switch. The **hats** daemon cannot obtain sufficient time to process CPU cycles during intensive operations (**df**, **find**, for example) and may be required to wait too long for a chance at the kernel lock. Often, more than five seconds will elapse before the **hatsd** can get a lock. The results are the invocation of the deadman switch and a system panic.

Solution

Determine what process is hogging CPU cycles on the system that panicked. Then attempt (in order) each of the following solutions that address this problem:

1. Tune the system using I/O pacing.
2. Increase the **syncd** frequency.
3. Change the Failure Detection Rate.

For instructions on these procedures, see the sections under Deadman Switch Causes a Node Failure earlier in this chapter.

Unplanned System Reboots Cause Fallover Attempt to Fail

Problem

Cluster nodes did not fallover after rebooting the system.

Solution

To prevent unplanned system reboots from disrupting a fallover in your cluster environment, all nodes in the cluster should either have the **Automatically REBOOT a system after a crash** field on the Change/Show Characteristics of Operating System SMIT panel set to **false**, or you should keep the IBM System p™ key in Secure mode during normal operation.

Both measures prevent a system from rebooting if the **shutdown** command is issued inadvertently. Without one of these measures in place, if an unplanned reboot occurs the activity against the disks on the rebooting node can prevent other nodes from successfully acquiring the disks.

Deleted or Extraneous Objects Appear in NetView Map

Problem

Previously deleted or extraneous object symbols appeared in the NetView map.

Solution

Rebuild the NetView database.

To rebuild the NetView database, perform the following steps on the NetView server:

1. Stop all NetView daemons: `/usr/OV/bin/ovstop -a`
2. Remove the database from the NetView server: `rm -rf /usr/OV/database/*`
3. Start the NetView object database: `/usr/OV/bin/ovstart ovwdb`
4. Restore the NetView/HAView fields: `/usr/OV/bin/ovw -fields`
5. Start all NetView daemons: `/usr/OV/bin/ovstart -a`

F1 Does not Display Help in SMIT Panels

Problem

Pressing F1 in SMIT panel does not display help.

Solution

Help can be displayed only if the LANG variable is set to one of the languages supported by HACMP, and if the associated HACMP message catalogs are installed. The languages supported by HACMP 5.4.1 are:

en_US	ja_JP
En_US	Ja_JP

To list the installed locales (the bsl LPPs), type:

```
locale -a
```

To list the active locale, type:

```
locale
```

Since the LANG environment variable determines the active locale, if LANG=en_US, the locale is en_US.

/usr/es/sbin/cluster/cl_event_summary.txt File (Event Summaries Display) Grows Too Large

Problem

In HACMP, event summaries are pulled from the **hacmp.out** file and stored in the **cl_event_summary.txt** file. This file continues to accumulate as hacmp.out cycles, and is not automatically truncated or replaced. Consequently, it can grow too large and crowd your **/usr** directory.

Solution

Clear event summaries periodically, using the **Problem Determination Tools > HACMP Log Viewing and Management > View/Save/Remove HACMP Event Summaries > Remove Event Summary History** option in SMIT.

View Event Summaries Does Not Display Resource Group Information as Expected

Problem

In HACMP, event summaries are pulled from the **hacmp.out** file and can be viewed using the **Problem Determination Tools > HACMP Log Viewing and Management > View/Save/Delete Event Summaries > View Event Summaries** option in SMIT. This display includes resource group status and location information at the end. The resource group information is gathered by **clRGinfo**, and may take extra time if the cluster is not running when running the **View Event Summaries** option.

Solution

clRGinfo displays resource group information more quickly when the cluster is running.

If the cluster is not running, wait a few minutes and the resource group information will eventually appear.

Application Monitor Problems

If you are running application monitors you may encounter occasional problems or situations in which you want to check the state or the configuration of a monitor. Here are some possible problems and ways to diagnose and act on them.

Problem 1

Checking the State of an Application Monitor. In some circumstances, it may not be clear whether an application monitor is currently running or not. To check on the state of an application monitor, run the following command:

```
ps -ef | grep <application server name> | grep clappmond
```

This command produces a long line of verbose output if the application is being monitored.

If there is no output, the application is not being monitored.

Solution 1

If the application monitor is not running, there may be a number of reasons, including

- No monitor has been configured for the application server
- The monitor has not started yet because the stabilization interval has not completed

- The monitor is in a suspended state
- The monitor was not configured properly
- An error has occurred.

Check to see that a monitor has been configured, the stabilization interval has passed, and the monitor has not been placed in a suspended state, before concluding that something is wrong.

If something is clearly wrong, reexamine the original configuration of the monitor in SMIT and reconfigure as needed.

Problem 2

Application Monitor Does Not Perform Specified Failure Action. The specified failure action does not occur even when an application has clearly failed.

Solution 2

Check the Restart Interval. If set too short, the Restart Counter may be reset to zero too quickly, resulting in an endless series of restart attempts and no other action taken.

Cluster Disk Replacement Process Fails

Problem

The disk replacement process fails while the **replacepv** command was running.

Solution

Be sure to delete the **/tmp/replacepv** directory, and attempt the replacement process again.

You can also try running the process on another disk.

Resource Group Unexpectedly Processed Serially

Problem

A resource group is unexpectedly processed serially even though you did not request it to be this way.

Solution

Check for the site policy that is specified for this resource group, and make sure it is set to **Ignore**. Then delete this resource group from the customized serial processing order list in SMIT and synchronize the cluster.

rg_move Event Processes Several Resource Groups at Once

Problem

In **hacmp.out**, you see that an **rg_move** event processes multiple non-concurrent resource groups in one operation.

Solution

This is the expected behavior. In clusters with dependencies, HACMP processes all resource groups upon **node_up** events, via **rg_move** events. During a single **rg_move** event, HACMP can process multiple non-concurrent resource groups within one event. For an example of the output, see the Processing in Clusters with Dependent Resource Groups or Sites section.

File System Fails to Unmount

Problem

A file system is not unmounted properly during an event such as when you stop cluster services with the option to bring resource groups offline.

Solution

One of the more common reasons for a file system to fail being unmounted when you stop cluster services with the option to bring resource groups offline is because the file system is busy. In order to unmount a file system successfully, no processes or users can be accessing it at the time. If a user or process is holding it, the file system will be "busy" and will not unmount.

The same issue may result if a file has been deleted but is still open.

The script to stop an application should also include a check to make sure that the shared file systems are not in use or deleted and in the open state. You can do this by using the **fuser** command. The script should use the **fuser** command to see what processes or users are accessing the file systems in question. The PIDs of these processes can then be acquired and killed. This will free the file system so it can be unmounted.

Refer to the AIX man pages for complete information on this command.

Dynamic Reconfiguration Sets a Lock

Problem

When attempting a DARE operation, an error message may be generated regarding a DARE lock if another DARE operation is in process, or if a previous DARE operation did not complete properly.

The error message suggests that one should take action to clear the lock if a DARE operation is not in process. "In process" here refers to another DARE operation that may have just been issued, but it also refers to any previous DARE operation that did not complete properly.

Solution

The first step is to examine the **/var/hacmp/log/hacmp.out** logs on the cluster nodes to determine the reason for the previous DARE failure. A **config_too_long** entry will likely appear in **hacmp.out** where an operation in an event script took too long to complete. If **hacmp.ou**t indicates that a script failed to complete due to some error, correct this problem and manually complete the remaining steps that are necessary to complete the event.

Run the HACMP SMIT **Problem Determination Tools** > **Recover from HACMP Script Failure** option. This should bring the nodes in the cluster to the next complete event state.

You can clear the DARE lock by selecting the HACMP SMIT option **Problem Determination Tools** > **Release Locks Set by Dynamic Configuration** if the HACMP SMIT **Recover from HACMP Script Failure** step did not do so.

WebSMIT Does Not “See” the Cluster

WebSMIT is designed to run on a single node. If that node goes down, WebSMIT will become unavailable. To increase availability, you can set up WebSMIT to run on multiple nodes. Since WebSMIT is retrieving and updating information from the HACMP cluster, that information should be available from all nodes in the cluster.

Typically, you will set up WebSMIT to be accessible from a cluster's internal network but not reachable from the Internet. If sites are configured, and WebSMIT is running on a node on a remote site, you must ensure HTTP connectivity to that node; it is not handled automatically by WebSMIT or HACMP. HTTPS/SSL is highly recommended for security.

Because WebSMIT runs on one node in the cluster, the functionality it provides and the information it displays directly correspond to the version of HACMP installed on that node. For HACMP 5.4.1 WebSMIT to work properly, you must have cluster services running on at least one node, and enable Javascript on the client.

Problems with WPAR-Enabled Resource Group

Problem

Resource Group fails to come online in a WPAR on a particular node.

Solution

1. Verify that the node in question is WPAR-capable. An AIX node with WPAR capability should have the **bos.wpars** fileset installed. If the node is not WPAR-capable, then the resource group will not run in the WPAR. Issue the following command to check if this fileset is installed:

```
lslpp -L "bos.wpars"
```

2. On the specified node, verify there is a WPAR with the same name as the WPAR-enabled resource group. Use the `lswpar <resource group name>` command to check this. If there is no WPAR with the specified name, create it using the **mkwpar** command. After creating a WPAR, make sure that all the user-defined scripts associated with the WPAR-enabled resource group are accessible within the WPAR.

3. Ensure that the file systems on the node are not full. If so, free up some disk space by moving some files to external storage.

4. Verify that the **rsh** service is enabled in the corresponding WPAR. This can be done as follows:

- Check that the **inetd** service is running in the WPAR by issuing the following command in the WPAR:

  ```
  lssrc -s inetd
  ```

  If the **inetd** service is not active, then start the service using the **startsrc** command.
- Make sure that **rsh** is listed as a known service in **/etc/inetd.conf** file in the WPAR.

Appendix A: Script Utilities

This appendix describes the utilities called by the event and startup scripts supplied with HACMP. These utilities are general-purpose tools that can be called from any script or from the AIX command line. The examples assume they are called from a script.

This appendix also includes the reference pages for Cluster Resource Group Information commands.

Highlighting

The following highlighting conventions are used in this appendix:

Bold	Identifies command words, keywords, files, directories, and other items whose actual names are predefined by the system.
Italics	Identifies parameters whose actual names or values are supplied by the user.
`Monospace`	Identifies examples of specific data values, examples of text similar to what you may see displayed, examples of program code similar to what you may write as a programmer, messages from the system, or information you should actually type.

Reading Syntax Diagrams

Usually, a command follows this syntax:

[]	Material within brackets is optional.
{}	Material within braces is required.
|	Indicates an alternative. Only one of the options can be chosen.
< >	Text within brackets is a variable.
...	Indicates that one or more of the kinds of parameters or objects preceding the ellipsis can be entered.

Note: Flags listed in syntax diagrams throughout this appendix are those recommended for use with the HACMP for AIX software. Flags used internally by SMIT are *not* listed.

Utilities

The script utilities are stored in the **/usr/es/sbin/cluster/events/utils** directory. The utilities described in this chapter are grouped in the following categories:

- Disk Utilities
- RS/6000 SP Utilities
- File System and Volume Group Utilities
- Logging Utilities
- Network Utilities
- Resource Group Move Utilities
- Emulation Utilities
- Security Utilities
- Start and Stop Tape Resource Utilities
- Cluster Resource Group Information Commands.

Disk Utilities

cl_disk_available

Syntax

```
cl_disk_available diskname ...
```

Description

Checks to see if a disk named as an argument is currently available to the system and if *not*, makes the disk available.

Parameters

diskname	List of one of more disks to be made available; for example, *hdisk1*.

Return Values

0	Successfully made the specified disk available.
1	Failed to make the specified disk available.
2	Incorrect or bad arguments were used.

cl_fs2disk

Syntax

```
cl_fs2disk [-lvip] mount_point
```

or

```
cl_fs2disk -g volume_group
```

where **-l** identifies and returns the logical volume, **-v** returns the volume group, **-i** returns the physical volume ID, **-p** returns the physical volume, and **-g** is the mount point of a file system (given a volume group).

Description

Checks the ODM for the specified logical volume, volume group, physical volume ID, and physical volume information.

Parameters

mount point	Mount point of file system to check.
volume group	Volume group to check.

Return Values

0	Successfully retrieved file system information.
1	Failed to retrieve file system information.

cl_get_disk_vg_fs_pvids

Syntax

```
cl_get_disk_vg_fs_pvids [filesystem_list volumegroup_list]
```

Description

Given file systems and/or volume groups, the function returns a list of the associated PVIDs.

Parameters

filesystem_list	The file systems to check.
volumegroup_list	The volume groups to check.

Return Values

0	Success.
1	Failure.
2	Invalid arguments.

cl_is_array

Syntax
```
cl_is_array diskname
```

Description
Checks to see if a disk is a READI disk array.

Parameters

diskname	Single disk to test; for example, *hdisk1*.

Return Values

0	Disk is a READI disk array.
1	Disk is *not* a READI disk array.
2	An error occurred.

cl_is_scsidisk

Syntax
```
cl_is_scsidisk diskname
```

Description
Determines if a disk is a SCSI disk.

Parameters

diskname	Single disk to test; for example, *hdisk1*.

Return Values

0	Disk is a SCSI disk.
1	Disk is *not* a SCSI disk.
2	An error occurred.

cl_raid_vg

Syntax
```
cl_raid_vg volume_group
```

Description
Checks to see if the volume group is comprised of RAID disk arrays.

Parameters

volume_group	Single volume group to check.

Return Values

0	Successfully identified a RAID volume group.
1	Could *not* identify a RAID volume group; volume group must be SSA.
2	An error occurred. Mixed volume group identified.

cl_scdiskreset

Syntax
```
cl_scdiskreset /dev/diskname ...
```

Description
Issues a reset (SCSI ioctl) to each SCSI disk named as an argument.

Parameters

/dev/diskname	List of one or more SCSI disks.

Return Values

0	All specified disks have been reset.
-1	No disks have been reset.
n	Number of disks successfully reset.

cl_scdiskrsv

Syntax
```
cl_scsidiskrsv /dev/diskname ...
```

Description
Reserves the specified SCSI disk.

Parameters

/dev/diskname	List of one or more SCSI disks.

Return Values

0	All specified disks have been reserved.
-1	No disks have been reserved.
n	Number of disks successfully reserved.

cl_sync_vgs

Syntax
```
cl_sync_vgs -b|f volume_group ...
```

Description
Attempts to synchronize a volume group by calling **syncvg** for the specified volume group.

Parameters

volume_group	Volume group list.
-b	Background sync.
-f	Foreground sync.

Return Values

0	Successfully started **syncvg** for all specified volume groups.
1	The **syncvg** of at least one of the specified volume groups failed.
2	No arguments were passed.

scdiskutil

Syntax
```
scdiskutil -t /dev/diskname
```

Description
Tests and clears any pending SCSI disk status.

Parameters

-t	Tests to see if a unit is ready.
/dev/diskname	Single SCSI disk.

Return Values

-1	An error occurred or no arguments were passed.
0	The disk is *not* reserved.
>0	The disk is reserved.

ssa_fence

Syntax
```
ssa_fence -e event pvid
```

Description
Fences a node in or out.

Additionally, this command also relies on environment variables; the first node up fences out all other nodes of the cluster regardless of their participation in the resource group.

If it is *not* the first node up, then the remote nodes fence in the node coming up. The node joining the cluster will *not* do anything.

If it is a **node_down** event, the remote nodes will fence out the node that is leaving. The node leaving the cluster will *not* do anything.

The last node going down clears the fence register.

Environment Variables

PRE_EVENT_MEMBERSHIP	Set by Cluster Manager.
POST_EVENT_MEMBERSHIP	Set by Cluster Manager.
EVENT_ON_NODE	Set by calling script.

Parameters

-e event	1=up; 2=down.
pvid	Physical volume ID on which fencing will occur.

Return Values

0	Success.
1	Failure. A problem occurred during execution. A message describing the problem is written to stderr and to the cluster log file.
2	Failure. Invalid number of arguments. A message describing the problem is written to stderr and to the cluster log file.

ssa_clear

Syntax

```
ssa_clear -x | -d pvid
```

Description

Clears or displays the contents of the fence register. If **-d** is used, a list of fenced out nodes will be displayed. If **-x** is used, the fence register will be cleared.

Note: This command exposes data integrity of a disk, by unconditionally clearing its fencing register. It requires adequate operator controls and warnings, and should *not* be included within any takeover script.

Return Values

0	Success.
1	Failure. A problem occurred during execution. A message describing the problem is written to stderr and to the cluster log file.
2	Failure. Invalid number of arguments. A message describing the problem is written to stderr and to the cluster log file.

ssa_clear_all

Syntax
```
ssa_clear_all pvid1, pvid2 ...
```

Description
Clears the fence register on multiple physical volumes.

Return Values

0	Success.
1	Failure. A problem occurred during execution. A message describing the problem is written to stderr and to the cluster log file.
2	Failure. Invalid number of arguments. A message describing the problem is written to stderr and to the cluster log file.

ssa_configure

Syntax
```
ssa_configure
```

Description
Assigns unique node IDs to all the nodes of the cluster. Then it configures and unconfigures all SSA pdisks and hdisks on *all* nodes thus activating SSA fencing. This command is called from the SMIT panel during the sync of a node environment. If this command fails for any reason, that node should be rebooted.

Return Values

0	Success.
1	Failure. A problem occurred during execution. A message describing the problem is written to stderr and to the cluster log file.

RS/6000 SP Utilities

cl_swap_HPS_IP_address

Syntax

```
cl_swap_HPS_IP_address [cascading rotating] [action] interface address
old_address netmask
```

Description

This script is used to specify an alias address to an SP Switch interface, or remove an alias address, during IP address takeover. Note that adapter swapping does *not* make sense for the SP Switch since all addresses are alias addresses on the same network interface.

Parameters

action	**acquire** or **release**
IP label behavior	**rotating/cascading**. Select **rotating** if an IP label should be placed on a boot interface; Select **cascading** if an IP label should be placed on a backup interface on a takeover node.
interface	The name of the interface.
address	new alias IP address
old_address	alias IP address you want to change
netmask	Netmask.

Return Values

0	Success.
1	The network interface could *not* be configured (using the **ifconfig** command) at the specified address.
2	Invalid syntax.

Examples

The following example replaces the alias *1.1.1.1* with *1.1.1.2*:

```
cl_swap_HPS_IP_address cascading acquire css0 1.1.1.2
    1.1.1.1 255.255.255.128
```

File System and Volume Group Utilities

The descriptions noted here apply to serial processing of resource groups. For a full explanation of parallel processing and JOB_TYPES, see Tracking Resource Group Parallel and Serial Processing in the hacmp.out File in Chapter 2: Using Cluster Log Files.

cl_activate_fs

Syntax

```
cl_activate_fs /filesystem_mount_point ...
```

Description

Mounts the file systems passed as arguments.

Parameters

/filesystem_mount_point	A list of one or more file systems to mount.

Return Values

0	All file systems named as arguments were either already mounted or were successfully mounted.
1	One or more file systems failed to **mount**.
2	No arguments were passed.

cl_activate_vgs

Syntax

```
cl_activate_vgs [-n] volume_group_to_activate ...
```

Description

Initiates a **varyonvg** of the volume groups passed as arguments.

Parameters

-n	Do *not* sync the volume group when **varyon** is called.
volume_group_to_activate	List of one of more volume groups to activate.

Return Values

0	All of the volume groups are successfully varied on.
1	The **varyonvg** of at least one volume group failed.
2	No arguments were passed.

cl_deactivate_fs

Syntax

```
cl_deactivate_fs /filesystem_mount_point ...
```

Description

Attempts to **unmount** any file system passed as an argument that is currently mounted.

Parameters

/filesystem_mount_point	List of one or more file systems to unmount.

Return Values

0	All file systems were successfully unmounted.
1	One or more file systems failed to unmount.
2	No arguments were passed.

cl_deactivate_vgs

Syntax

```
cl_deactivate_vgs volume_group_to_deactivate ...
```

Description

Initiates a **varyoffvg** of any volume group that is currently varied on and that was passed as an argument.

Parameters

volume_group_to_deactivate	List of one or more volume groups to vary off.

Return Values

0	All of the volume groups are successfully varied off.
1	The **varyoffvg** of at least one volume group failed.
2	No arguments were passed.

cl_nfskill

Syntax

```
cl_nfskill [-k] [-t] [-u] directory ...
```

Description

Lists the process numbers of local processes using the specified NFS directory.

Find and kill processes that are executables fetched from the NFS-mounted file system. Only the root user can kill a process of another user.

If you specify the **-t** flag, all processes that have certain NFS module names within their stack will be killed.

WARNING: When using the **-t** flag it is *not* possible to tell which NFS file system the process is related to. This could result in killing processes that belong to NFS-mounted file systems other than those that are cross-mounted from another HACMP node and under HACMP control. This could also mean that the processes found could be related to file systems under HACMP control but *not* part of the current resources being taken. This flag should therefore be used with caution and only if you know you have a specific problem with unmounting the NFS file systems.

To help to control this, the **cl_deactivate_nfs** script contains the normal calls to **cl_nfskill** with the **-k** and **-u** flags and commented calls using the **-t** flag as well. If you use the **-t** flag, you should uncomment those calls and comment the original calls.

Parameters

-k	Sends the SIGKILL signal to each local process,
-u	Provides the login name for local processes in parentheses after the process number.
-t	Finds and kills processes that are just opening on NFS file systems.
directory	Lists of one or more NFS directories to check.

Return Values

None.

Logging Utilities

cl_log

Syntax
```
cl_log message_id default_message variables
```

Description
Logs messages to **syslog** and standard error.

Parameters

message_id	Message ID for the messages to be logged.
default_message	Default message to be logged.
variables	List of one or more variables to be logged.

Return Values

0	Successfully logged messages to **syslog** and standard error.
2	No arguments were passed.

cl_echo

Syntax
```
cl_echo message_id default_message variables
```

Description
Logs messages to standard error.

Parameters

message_id	Message ID for the messages to be displayed.
default_message	Default message to be displayed.
variables	List of one or more variables to be displayed.

Return Values

0	Successfully displayed messages to stdout.
2	No arguments were passed.

Network Utilities

cl_swap_HW_address

Syntax

```
cl_swap_HW_address address interface
```

Description

Checks to see if an alternate hardware address is specified for the address passed as the first argument. If so, it assigns the hardware address specified to the network interface.

Parameters

address	Interface address or IP label.
interface	Interface name (for example, *en0* or *tr0*).

Return Values

0	Successfully assigned the specified hardware address to a network interface.
1	Could *not* assign the specified hardware address to a network interface.
2	Wrong number of arguments were passed.

Note: This utility is used during adapter_swap and IP address takeover.

cl_swap_IP_address

Syntax

```
cl_swap_IP_address cascading/rotating acquire/release interface
new_address old_address netmask

cl_swap_IP_address swap_adapter swap interface1 address1 interface2
address2 netmask
```

Description

This routine is used during adapter_swap and IP address takeover.

In the first form, the routine sets the specified interface to the specified address:

```
cl_swap_IP_address rotating acquire en0 1.1.1.1 255.255.255.128
```

In the second form, the routine sets two interfaces in a single call. An example where this is required is the case of swapping two interfaces.

```
cl_swap_IP_address swap-adapter swap en0 1.1.1.1 en1 2.2.2.2
255.255.255.128
```

Parameters

interface	Interface name
address	IP address
IP label behavior	**rotating/cascading**. Select **rotating** if an IP label should be placed on a boot interface; select **cascading** if an IP label should be placed on a backup interface on a takeover node.
netmask	Network mask. Must be in decimal format.

Return Values

0	Successfully swapped IP addresses.
1	**ifconfig** failed.
2	Wrong or incorrect number of arguments.

This utility is used for swapping the IP address of either a standby network interface with a local service network interface (called adapter swapping), or a standby network interface with a remote service network interface (called masquerading). For masquerading, the **cl_swap_IP_address** routine should sometimes be called *before* processes are stopped, and sometimes *after* processes are stopped. This is application dependent. Some applications respond better if they shutdown *before* the network connection is broken, and some respond better if the network connection is closed first.

cl_unswap_HW_address

Syntax

```
cl_unswap_HW_address interface
```

Description

Script used during adapter_swap and IP address takeover. It restores a network interface to its boot address.

Parameters

interface	Interface name (for example, *en0* or *tr0*).

Return Values

0	Success.
1	Failure.
2	Invalid parameters.

Resource Group Move Utilities

clRGmove

The **clRGmove** utility is stored in the **/usr/es/sbin/cluster/utilities** directory.

Syntax

The general syntax for migrating, starting, or stopping a resource group dynamically from the command line is this:

```
clRGmove -g <groupname> [-n <nodename> | -r | -a] [-m | -u | -d] [-s
true|false ] [-p] [-i]
```

Description

This utility communicates with the Cluster Manager to queue an **rg_move** event to bring a specified resource group offline or online, or to move a resource group to a different node. This utility provides the command line interface to the Resource Group Migration functionality, which can be accessed through the SMIT **System Management (C-SPOC)** panel. To move specific resource groups to the specified location or state, use the **System Management (C-SPOC)** > **HACMP Resource Group and Application Management > Move a Resource Group to another Node/Site** SMIT menu, or the **clRGmove** command. See the man page for the **clRGmove** command.

You can also use this command from the command line, or include it in the pre- and post-event scripts.

Parameters

-a	Use this flag for concurrent resource groups only. This flag is interpreted as **all nodes in the resource group** when bringing the concurrent resource group offline or online. To bring a concurrent resource group online or offline on a *single node*, use the **-n** flag.
-d	Use this flag to bring the resource group offline. Cannot be used with **-m** or **-u** flags.
-g <*groupname*>	The name of the resource group to move.
-i	Displays the locations and states of all resource groups in the cluster after the migration has completed by calling the **clRGinfo** command.
-m	Use this flag to move one or more resource groups from their current node to a specified destination node. Cannot be used with **-d** or **-u** flags. Example: `clRGmove -g "rgA, rgB, rgC" -n nodeB -m`

-n <*nodename*>	The name of the node to which the resource group will be moved. For a non-concurrent resource group this flag can only be used when bringing a resource group online or moving a resource group to another node. For a concurrent resource group this flag can be used to bring a resource group online or offline on a single node. Cannot be used with **-r** or **-a** flags.
-p	Use this flag to show the temporal changes in the resource group behavior that occurred because of the resource group migration utility.
-r	This flag can only be used when bringing a non-concurrent resource group online or moving a non-concurrent resource group to another node. If this flag is specified, the command uses the highest priority node that is available as the destination node to which the group will be moved. Cannot be used with **-n** or **-a** flags.
-s true | false	Use this flag to specify actions on the primary or secondary instance of a resource group (if sites are defined). With this flag, you can take the primary or the secondary instance of the resource group offline, online or move it to another node within the same site. **- s true** specifies actions on the secondary instance of a resource group. **-s false** specifies actions on the primary instance of a resource group. Use this flag with **−r**, **-d**, **-u**, and **-m** flags.
-u	Use this flag to bring the resource group online. Cannot be used with **-m** or **-d** flags.

Repeat this syntax on the command line for each resource group you want to migrate.

Return Values

0	Success.
1	Failure.

Emulation Utilities

Emulation utilities are found in the /**usr/es/sbin/cluster/events/emulate/driver** directory.

cl_emulate

Syntax

```
cl_emulate -e node_up -n nodename
cl_emulate -e node_down -n nodename {f|g|t}
cl_emulate -e network_up -w networkname -n nodename
cl_emulate -e network_down -w networkname -n nodename
cl_emulate -e join_standby -n nodename -a ip_label
cl_emulate -e fail_standby -n nodename -a ip_label
cl_emulate -e swap_adapter -n nodename -w networkname -a ip_label
              -d ip_label
```

Description

Emulates a specific cluster event and outputs the result of the emulation. The output is shown on the screen as the emulation runs, and is saved to an output file on the node from which the emulation was executed.

The Event Emulation utility does *not* run customized scripts such as pre- and post- event scripts. In the output file the script is echoed and the syntax is checked, so you can predict possible errors in the script. However, if customized scripts exist, the outcome of running the actual event may differ from the outcome of the emulation.

When emulating an event that contains a customized script, the Event Emulator uses the **ksh** flags **-n** and **-v**. The **-n** flag reads commands and checks them for syntax errors, but does *not* execute them. The **-v** flag indicates verbose mode. When writing customized scripts that may be accessed during an emulation, be aware that the other **ksh** flags may *not* be compatible with the **-n** flag and may cause unpredictable results during the emulation. See the **ksh** man page for flag descriptions.

You can run only one instance of an event emulation at a time. If you attempt to start an emulation while an emulation is already running on a cluster, the integrity of the output cannot be guaranteed.

Parameters

-e *eventname*	The name of the event to **emulate: node_up**, **node_down**, **network_up**, **network_down**, **join standby**, **fail standby**, **swap adapter**.
-n *nodename*	The node name used in the emulation.
-f	Emulates stopping cluster services with the option to place resource groups in an UNMANAGED state. Cluster daemons terminate without running any local procedures.
-g	Emulates stopping cluster services with the option to bring resource groups offline.
-t	Emulates stopping cluster services with the option to move resource groups to another node.
-w *networkname*	The network name used in the emulation.
-a *ip_label*	The standby network interface address with which to switch.
-d *ip_label*	The service network interface to fail.

Return Values

0	Success.
1	Failure.

Note: The **cldare** command also provides an emulation feature for dynamic reconfiguration events.

Security Utilities

HACMP security utilities include kerberos setup utilities.

Kerberos Setup Utilities

To simplify and automate the process of configuring Kerberos security on the SP, two scripts for setting up Kerberos service principals are provided with HACMP:

- **cl_setup_kerberos**—extracts the HACMP network interface labels from an already configured node and creates a file, **cl_krb_service**, that contains all of the HACMP network interface labels and additional format information required by the **add_principal** Kerberos setup utility. Also creates the **cl_adapters** file that contains a list of the network interfaces required to extract the service principals from the authentication database.
- **cl_ext_krb**—prompts the user to enter the Kerberos password to be used for the new principals, and uses this password to update the **cl_krb_service** file. Checks for a valid **.k** file and alerts the user if one does not exist. Once a valid **.k** file is found, the **cl_ext_krb** script runs the **add_principal** utility to add all the network interface labels from the **cl_krb_service** file into the authentication database; extracts the service principals and places them in a new Kerberos services file, **cl_krb-srvtab**; creates the **cl_klogin** file that contains additional entries required by the **.klogin** file; updates the **.klogin** file on the control workstation and all nodes in the cluster; and concatenates the **cl_krb-srvtab** file to each node's **/etc/krb-srvtab** file.

Start and Stop Tape Resource Utilities

The following sample scripts are supplied with the software.

tape_resource_start_example

Syntax

```
tape_resource_start_example
```

Description

Rewinds a highly available tape resource.

Parameters

none

Return Values

0	Successfully started the tape resource.
1	Failure.
2	Usage error.

tape_resource_stop_example

Syntax

```
tape_resource_stop_example
```

Description

Rewinds the highly available tape resource.

Parameters

None.

Return Values

0	Successfully stopped the tape resource.
1	Failure.
2	Usage error.

Cluster Resource Group Information Commands

The following commands are available for use by scripts or for execution from the command line:

- **clRMupdate**
- **clRGinfo**.

HACMP event scripts use the **clRMupdate** command to notify the Cluster Manager that it should process an event. It is *not* documented for end users; it should only be used in consultation with IBM support personnel.

Users or scripts can execute the **clRGinfo** command to get information about resource group status and location.

clRGinfo

Syntax

```
clRGinfo [-a] [-h] [-v] [-s| -c] | [-p] [-t] [-d][groupname1]
[groupname2] ...
```

Description

Use the **clRGinfo** command to display the location and state of all resource groups.

If **clRGinfo** cannot communicate with the Cluster Manager on the local node, it attempts to find a cluster node with the Cluster Manager running, from which resource group information may be retrieved. If **clRGinfo** fails to find at least one node with the Cluster Manager running, HACMP displays an error message.

```
clRGinfo: Resource Manager daemon is unavailable
```

Parameters

s or **c**	Displays output in colon (shortened) format.
a	Displays the pre-event and the post- event node locations of the resource group. (Recommended for use in pre- and post-event scripts in clusters with dependent resource groups).
d	Displays the name of the server that provided the information for the command.
p	Displays the node that temporarily has the highest priority for this instance as well as the state for the primary and secondary instances of the resource group. The command shows information about those resource groups whose locations were temporally changed because of the resource group migration utility.
t	Collects information only from the local node and displays the delayed fallback timer and the settling time settings for resource groups on the local node. Note: This flag can be used *only* if the Cluster Manager is running on the local node.
h	Displays the usage message.
v	Displays the verbose output with the startup, fallover and fallback policies for resource groups

Return Values:

0	Success
1	Operation could *not* be performed

Examples of clRGinfo Output

clRGinfo

```
$ /usr/es/sbin/cluster/utilities/clRGinfo
-----------------------------------------------------------------------------
Group Name        Group State     Node           Node State
-----------------------------------------------------------------------------
Group1            ONLINE          merry          ONLINE
                                  samwise        OFFLINE

Group2            ONLINE          merry          ONLINE
```

If you run the **clRGinfo** command with sites configured, that information is displayed as in the following example:

```
$ /usr/es/sbin/cluster/utilities/clRGinfo
-----------------------------------------------------------------------------
Group Name        Group State     Node           Node State
-----------------------------------------------------------------------------
Colors            ONLINE          white@Site1    ONLINE
                                  amber@Site1    OFFLINE
                                   yellow@Site1   OFFLINE

                                   navy@Site2     ONLINE_SECONDARY
                                   ecru@Site2     OFFLINE
samwise          ONLINE
```

clRGinfo -c -p

If you run the **clRGinfo -c -p** command, it lists the output in a colon separated format and the parameter indicating status and location of the resource groups

Possible States of a Resource Group

```
ONLINE
OFFLINE
OFFLINE Unmet dependencies
OFFLINE User requested
UNKNOWN
ACQUIRING
RELEASING
ERROR
TEMPORARY ERROR
ONLINE SECONDARY
ONLINE PEER
ACQUIRING SECONDARY
RELEASING SECONDARY
ACQUIRING PEER
RELEASING PEER
ERROR SECONDARY
TEMPORARY ERROR SECONDARY
```

clRGinfo -a

The **clRGinfo -a** command lets you know the pre-event location and the post-event location of a particular resource group. Because HACMP performs these calculations at event startup, this information will be available in pre-event scripts (such as a pre-event script to node_up), on all nodes in the cluster, regardless of whether the node where it is run takes any action on a particular resource group.

Note: **clRGinfo - a** provides meaningful output *only* if you run it while a cluster event is being processed.

- In this example, the resource group A is moving from the offline state to the online state on node B. The pre-event location is left blank, the post-event location is Node B:

```
:rg_move[112] /usr/es/sbin/cluster/utilities/clRGinfo -a
-----------------------------------------------------
Group Name Resource Group Movement
-----------------------------------------------------
rgA           PRIMARY=":nodeB"
```

- In this example, the resource group B is moving from Node B to the offline state. The pre-event location is node B, the post-event location is left blank:

```
:rg_move[112] /usr/es/sbin/cluster/utilities/clRGinfo -a
-----------------------------------------------------
Group Name Resource Group Movement
-----------------------------------------------------
rgB           PRIMARY="nodeB:"
```

- In this example, the resource group C is moving from Node A to Node B. The pre-event location is node A, the post-event location is node B:

```
:rg_move[112] /usr/es/sbin/cluster/utilities/clRGinfo -a
-----------------------------------------------------
Group Name Resource Group Movement
-----------------------------------------------------
rgC           PRIMARY="nodeA:nodeB"
```

- In this example with sites, the primary instance of resource group C is moving from Node A to Node B, and the secondary instance stays on node C:

```
:rg_move[112] /usr/es/sbin/cluster/utilities/clRGinfo -a
-----------------------------------------------------
Group Name Resource Group Movement
-----------------------------------------------------
rgC           PRIMARY="nodeA:nodeB"
              SECONDARY="nodeC:nodeC"
```

- With concurrent resource groups, the output indicates each node from which a resource group is moving online or offline. In the following example, both nodes release the resource group:

```
:rg_move[112] /usr/es/sbin/cluster/utilities/clRGinfo -a
-----------------------------------------------------
Group Name Resource Group Movement
-----------------------------------------------------
rgA   "nodeA:"
rgA   "nodeB:"
```

clRGinfo -p

The **clRGinfo -p** command displays the node that temporarily has the highest priority for this instance as well as the state for the primary and secondary instances of the resource group. The command shows information about those resource groups whose locations were temporally changed because of the resource group migration utility.

```
$ /usr/es/sbin/cluster/utilities/clRGinfo -p

Cluster Name: TestCluster

Resource Group Name: Parent
Primary instance(s):
The following node temporarily has the highest priority for this
instance:
```

```
user-requested rg_move performed on Wed Dec 31 19:00:00 1969

Node                          State
----------------------------- ---------------
node3@s2                    OFFLINE
node2@s1                    ONLINE
node1@s0                    OFFLINE

Resource Group Name: Child
Node                          State
----------------------------- ---------------
node3@s2                    ONLINE
node2@s1                    OFFLINE
node1@s0                    OFFLINE

----------------------------- ---------------
```

clRGinfo -p -t

The **clRGinfo -p -t** command displays the node that temporarily has the highest priority for this instance and a resource group's active timers:

```
/usr/es/sbin/cluster/utilities/clRGinfo -p -t
Cluster Name: MyTestCluster

Resource Group Name: Parent
Primary instance(s):
The following node temporarily has the highest priority for this
instance:
node4, user-requested rg_move performed on Fri Jan 27 15:01:18 2006
Node           Primary State    Secondary StateDelayed Timers
------------------------------ ---------------- -------------------
node1@siteA    OFFLINE          ONLINE SECONDARY
node2@siteA    OFFLINE          OFFLINE
node3@siteB    OFFLINE          OFFLINE
node4@siteB    ONLINE           OFFLINE

Resource Group Name: Child
Node           State            Delayed Timers
------------------------------ --------------- -------------------
node2          ONLINE
node1          OFFLINE
node4          OFFLINE
node3          OFFLINE
```

clRGinfo -s

```
$ /usr/es/sbin/cluster/utilities/clRGinfo -s
Group1:ONLINE:merry:OHN:FNPN:FBHPN:ignore: : :
Group1:OFFLINE:samwise:OHN:FNPN:FBHPN:ignore: : :
Group2:ONLINE:merry:OAAN:BO:NFB:ignore: : :
Group2:ONLINE:samwise:OAAN:BO:NFB:ignore: : :
```

The **-s** flag prints the output in the following order:

```
RGName:state:node:type:startup:fallover:fallback:site:POL:POL_SEC:
fallbackTime:settlingTime

$ /usr/es/sbin/cluster/utilities/clRGinfo -s
Group1:ONLINE:white::ONLINE:OHN:FNPN:FBHPN:PPS: : : :ONLINE:Site1:
Group1:OFFLINE:amber::OFFLINE:OHN:FNPN:FBHPN:PPS: : : :ONLINE:Site1 :
Group1:ONLINE:yellow::ONLINE:OAAN:BO:NFB:PPS: : : :ONLINE:Site1:
Group1:ONLINE:navy::ONLINE:OAAN:BO:NFB:PPS: : : :ONLINE:Site2:
Group1:ONLINE:ecru::ONLINE:OAAN:BO:NFB:PPS: : : :ONLINE:Site2:
```

where the resource group startup fallover and fallback preferences are abbreviated as follows:

- Resource group startup policies:
  - OHN: Online On Home Node Only
  - OFAN: Online On First Available Node
  - OUDP: Online Using Distribution Policy
  - OAAN: Online On All Available Nodes
- Resource group fallover policies:
  - FNPN: fallover To Next Priority Node In The List
  - FUDNP: fallover Using Dynamic Node Priority
  - BO: Bring Offline (On Error Node Only)
- Resource group fallback policies:
  - FHPN: Fallback To Higher Priority Node In The List
  - NFB: Never Fallback.
- Resource group's intersite policies:
  - ignore: ignore
  - OES: Online On Either Site
  - OBS: Online Both Sites
  - PPS: Prefer Primary Site

If an attribute is *not* available for a resource group, the command displays a colon and a blank instead of the attribute.

clRGinfo -v

```
$ /usr/es/sbin/cluster/utilities/clRGinfo -v

Cluster Name: myCLuster

Resource Group Name: Group1
Startup Policy: Online On Home Node Only
fallover Policy: fallover To Next Priority Node In The List
Fallback Policy: Fallback To Higher Priority Node In The List
Site Policy: ignore

Node          State
--------------- ---------------
merry            ONLINE
samwise          OFFLINE

Resource Group Name: Group2
Startup Policy: Online On All Available Nodes
fallover Policy: Bring Offline (On Error Node Only)
Fallback Policy: Never Fallback
Site Policy: ignore


Node             State
--------------- ---------------
merry            ONLINE
samwise          ONLINE
```

Appendix B: Command Execution Language Guide

This appendix describes how you can use the HACMP for AIX Command Execution Language (CEL) to create additional Cluster Single Point of Control (C-SPOC) commands for your cluster. It also explains CEL constructs and how to convert a command's "execution plan" written in CEL into a Korn shell (**ksh**) C-SPOC script.

You should be familiar with Korn shell programming (and with programming concepts in general) before attempting to create C-SPOC commands.

Overview

CEL is a programming language that lets you integrate the **dsh** command's distributed functionality into each C-SPOC script the CEL preprocessor **(celpp**) generates. When you invoke a C-SPOC script from a single cluster node to perform an administrative task, the script is automatically executed on all nodes in the cluster. Without C-SPOC's distributed functionality, you must execute each administrative task separately on each cluster node, which can lead to inconsistent node states within the cluster.

Appendix C: HACMP for AIX Commands in the *Administration Guide* provides a list of all C-SPOC commands provided with the HACMP for AIX software.

Creating C-SPOC Commands

C-SPOC commands are written as execution plans in CEL. Each plan contains constructs to handle one or more underlying AIX tasks (a command, executable, or script) with a minimum of user input. An execution plan becomes a C-SPOC command when the **/usr/es/sbin/cluster/utilities/celpp** utility converts it into a *cluster aware* **ksh** script, meaning the script uses the C-SPOC distributed mechanism—the C-SPOC Execution Engine—to execute the underlying AIX commands on cluster nodes to complete the defined tasks.

C-SPOC commands provide a means for controlling the execution of specific tasks on cluster nodes, collecting status and log information in the **/var/hacmp/log/cspoc.log** file, and responding to errors generated by its script. Each command corresponds to an existing, underlying AIX system administration task that lets you maintain user accounts, maintain shared Logical Volume Manager (LVM) components, or control HACMP cluster services on a cluster-wide basis.

To create a C-SPOC command:

1. Write an execution plan for the command.
2. Run **celpp** to convert the plan (**.cel** file) into a **ksh** script and make the script executable.
3. Store the script in a user-defined directory.

The following sections describe each step.

Writing an Execution Plan

In a C-SPOC cluster environment, various commands let you manage user and group accounts, logical volumes, and cluster services. You can expand this command set by creating new commands for specific tasks that meet your administrative needs.

To create a new command, you first write an execution plan for the command. The execution plan contains **ksh** and CEL constructs (statements and clauses) that together describe the tasks the script will perform. CEL constructs are shown in bold in the following pseudo execution plan example. Execution plans you create should be similar in format and content to this example.

For an explanation of each construct and to see an actual execution plan for the **cl_chuser** command see the CEL Constructs section.

```
#####################################################################
#Other CEL scripts may be included. This one defines routines used
#by all C-SPOC scripts (e.g. initialization, verification, logging,
#etc.)
%include cl_path.cel
%include cl_init.cel
#The following variables must be defined when including cl_init.cel in
an execution plan:

_CSPOC_OPT_STR=""       # Define valid C-SPOC option flags
_OPT_STR=""             # Define valid option flags for the generated
script
_USAGE=""               # Define the usage for the generated script
%define ERROR_CODE 2
#An error that is not handled within a try statement using %except
#can be handled using a global %others statement.

%others
  print "ERROR: Command failed!"
%end
#Each plan must contain a try statement to execute commands across
#nodes in the cluster. A %try_parallel executes commands on all
#nodes simultaneously. Note that only a single line ksh statement
#is allowed after a %try_parallel or %try_serial.

%try_parallel
  ls -l $*
  %except ERROR_CODE
    print "ERROR: Unable to open: $*"
  %end
%end
#####################################################################
```

For a description of option string variables used in the preceding example, refer to the **cl_init.cel** file in the **/usr/es/sbin/cluster/samples/cspoc** directory. The **cl_init.cel** file provides examples of functionality required in any execution plan you create; it should be included in each **.cel** file.

The initialization and verification routines in the **cl_init.cel** file provide the following functionality:

- Get a list of nodes in the cluster.

- Get a list of target nodes for command execution.
- Determine nodes associated with any resource groups specified.
- Process and implement the standard C-SPOC flags (**-f**, **-n**, and **-g**).
- Validate option strings. Requires several environment variables to be set within the plan (**_OPT_STR**, **CSPOC_OPT_STR**, and **_USAGE**).
- Save log file entries upon command termination.
- Perform C-SPOC verification as follows:
  - Ensure **dsh** is available in **$PATH**
  - Check the HACMP version on all nodes.

The **cl_path .cel** file sets the PATH variable so that the C-SPOC and HACMP functions can be found at runtime. This is essential for execution plans that make use of any HACMP command, or the C-SPOC try_serial or try_parallel operations.

Encoding and Decoding Command-Line Arguments

If the command execution plans you create require that you pass additional arguments (or perform additional processing to the existing command-line arguments), you must use the C-SPOC **clencodearg** and **cldecodearg** programs located in **/usr/es/sbin/cluster/cspoc** directory to code and decode a list of arguments. This ensures that the CEL execution engine handles the embedded spacing and quotes within arguments across multiple invocations of the shell.

For example, in the following command, the argument “John Smith” contains an embedded space:

```
chuser gecos="John Smith" jsmith
```

Thus, you should encode all command-line arguments to C-SPOC commands unless they begin with a dash. Arguments that begin with a dash are generally command-line flags and do *not* contain spaces or quotes. If a string that begins with a dash is passed to the **clencodearg** or **cldecodearg** program, the string is passed through without being encoded or decoded.

The following script fragments shows how to encode and decode a list of arguments.

To encode a list of args:

```
ENCODED_ARGS=""
for ARG in "$@"
do
  ENCODED_ARGS="$ENCODED_ARGS $(print ARG | clencodearg)"
done
To decode a list of args:
UNENCODED_ARGS=""
for ARG in $ENCODED_ARGS
do
  UNENCODED_ARGS="$UNENCODED_ARGS $(print $ARG | cldecodearg)"
done
```

WARNING: You must use the **clencodearg** or **cldecodearg** program to process command-line arguments that are to be passed to any commands contained within a **%try_serial** or **%try_parallel** statement because the C-SPOC Execution Engine (**cdsh**) tries to decode all command-line arguments before executing the command. See CEL Constructs for more information about **%try_serial** and **%try_parallel** statements.

WARNING: Any arguments obtained from the environment variables set by routines within **cl_init.cel**, such as **_getopts()**, will have been encoded. If a command contained within a **%try_serial** or **%try_parallel** statement includes arguments generated within the execution plan, then they must first be encoded. For example, most execution plans pass command-line arguments as follows:

```
%try_parallel
   chuser $_CMD_ARGS
%end
```

The following CEL plan uses the **clencodearg** and **cldecodearg** programs.

Example CEL Plan

```
#Initialize C-SPOC variables required by cl_init.cel

  _CMD_NAME=$(basename $0) # Specify the name of this script.
  _CSPOC_OPT_STR="d:f"     # Specify valid C-SPOC option flags.
  _OPT_STR="+2"            # Specify valid AIX command option flags.

#Specify a Usage statement for this script.

  _USAGE="USAGE: cl_chuser [-cspoc -f] Attr=Value...Username"
#Initialize variables local to this script. The default return code
#is '0' for success.

  RETCODE=0
#Include the required C-SPOC Initialization and Verification
#Routines

   %include cl_path.cel
   %include cl_init.cel
#This define makes the following exception clause more readable.

  %define USER_NOT_FOUND 2
#Get the username, which is the last command line arg and perform
#some crude checks for validity. Note: This is an example of
#decoding args within a script.

  user=${_CMD_ARGS##*[     ]} # The []'s contain a tab & a space!
  case $user in
  -*|"")
        print "$_USAGE"
        exit 2
        ;;
  esac
#Since cl_init.cel has encoded all the args we must decode it to
# use it.

    Duser=$(print $user | cldecodearg)
#Construct a Gecos field based on the 411 entry for the username.
```

```
#Note: 411 is a local script that prints the following
# tab-separated fields: username   Firstname Lastname  Work_phone
# Home_phone

#This plan cuts the "Firstname Lastname" to put into the Gecos
#field of /etc/passwd
```

```
  GECOS=$(/usr/local/bin/411 $Duser | cut -d'     ' -f2)
#Construct a new set of command line args. Note the following:
# 1) We put the 'gecos=' arg just before the username so that it
#    will supercede any that may have been specified on the
#    command line.
# 2) We must encode the 'gecos=' arg explicitly.
# 3) This is an example of encoding args specified inside a script.

 NEW_CMD_ARGS=${_CMD_ARGS%[      ]*} # []'s contain a tab & a space
 NEW_CMD_ARGS="$NEW_CMD_ARGS $(print gecos="$GECOS" | HR>
clencodearg) $user"

#Perform a check that the username exists on all nodes. The check
#is not performed when the C-SPOC force flag is specified.

  if [[ -z "${_SPOC_FORCE}" ]]
  then

#Check if user exists across all cluster nodes

  %try_parallel _NODE _TARGET_NODES silent_err silent_out
    lsuser $user
    %except USER_NOT_FOUND
      print "${_CMD_NAME}: User ${Duser} does "
      print "not exist on node ${_NODE}" >&2"
      RETCODE=1
    %end
  %end
  fi
#If the username does not exist on a node then exit immediately

  if [[ ${RETCODE} -ne 0 ]]
  then
        exit ${RETCODE}
  fi

#Execute the chuser command in parallel on all nodes in the #cluster.

  %try_parallel _NODE _TARGET_NODES
        chuser $NEW_CMD_ARGS
        %others
          # If chuser returned an error on any node
          # set the return code to '1' to indicate
          # that one or more nodes failed.
             RETCODE=1
        %end
  %end

#Exit with the appropriate value.

  exit ${RETCODE}
```

CEL Constructs

You can use the following CEL constructs (statements or clauses) in a command's execution plan. All C-SPOC commands must contain the **%include** statement, a **%try_parallel** or **%try_serial** statement, and an **%end** statement.

The **%include** statement is used to access **ksh** libraries within the **cl_init.cel** file. These libraries make C-SPOC commands "cluster aware." The **%except** and **%others** clauses are used typically for error handling in response to a command's execution on one or more cluster nodes.

%define statement:

```
%define key value
```

For improved readability, the **%define** statement (keyword) is used to provide descriptive names for **error_id** values in **%except** clauses. The **error_id** given in the define statement is inserted in place of the given ID in any subsequent **%except** clauses.

%include statement:

```
%include filename
```

The **%include** statement allows a copy of a file to be included in an execution plan, which means that common code can be shared among execution plans. The **%include** statement also allows C-SPOC commands with similar algorithms to share the same plans. Using **%include**, CEL statements can be used within library routines included in any execution plan.

%try_parallel statement:

```
%try_parallel node nodelist [silent_err] [silent_out]
      ksh statement
      [ %except clause ... ]
[%end | %end_p ]
```

The **%try_parallel** statement executes the enclosed **ksh** statement simultaneously across all nodes in the nodelist. It waits until a command completes its execution on all cluster nodes before checking for errors.

The **%try_parallel** statement is equivalent to the following pseudo-code:

```
for each node in nodelist
    execute ksh statement in the background on each node
end
wait for all background execution to complete on all nodes
for each node in nodelist
    check status for each node and execute %except
  clause(s)
end
```

%try_serial statement:

```
%try_serial node nodelist [silent_err] [silent_out]
      ksh statement
      [ %except clause ... ]
      [%others clause]
[%end | %end_s ]
```

The **%try_serial** statement executes the enclosed **ksh** statement consecutively on each node specified in the nodelist. The command must complete before **%try_serial** continues its execution on the next node. All **%try_serial** statements sequentially check for errors after a command completes on each node in the list.

The **%try_serial** statement is equivalent to the following pseudo-code:

```
for each node in nodelist
        execute ksh statement in the on each node
    end
    for each node in nodelist
        check status for each node and execute %except
   clause(s)
    end
```

Note: Any non-flag arguments (those *not* preceded by a dash) used inside a **try** statement must be encoded using the **/usr/es/sbin/cluster/cspoc/clencodearg** utility; otherwise, arguments will be decoded incorrectly.

%except clause:

```
%except error_id
      ksh code
[ %end | %end_e ]
```

The **%except** clause is used for error checking on a per-node basis, and for the execution of the **ksh** statement contained within a **%try_parallel** or **%try_serial** statement. If the **ksh** statement fails or times out, the **%except** clauses define actions to take for each return code value.

If an **error_id** is defined using the **%define** statement, the return code of the **ksh** statement (the status code set by a command's execution on a particular node) is compared to the **error_id** value. If a match is found, the **ksh** code within the **%except** statement is executed.

If the **%others** clause is used within the **%except** clause, the **ksh** code within the **%except** clause is executed only if the return code value did *not* match an **error_id** defined in previous **%except** clauses.

An **%except** clause can be defined within or outside a **try** statement. If it is defined within a **try** statement, the clause's scope is local to the **%try_parallel** or **%try_serial** statement.

If the **%except** clause is defined outside a **try** statement, it is used in any subsequent **%try_serial** or **%try_parallel** statements that do *not* already provide one for the specified **error_id** or **%others** clause. In this case, the **%except** clause's scope is global until an **%unexcept** statement is found. Global **%except** clauses are used to simplify and standardize repetitive types of error handling.

%stop_try clause:

```
%except error_id
      ksh code
      [ %stop_try clause ]
      ksh code
[ %end | %end_e ]
```

The **%stop_try** clause can be used within an **%except** clause; it causes an exit from the enclosing **try** statement. This statement has the effect of a break statement in other languages.

The **%stop_try** clause has a different effect that depends on whether it is defined with a **%try_parallel** versus a **%try_serial** statement.

In a **%try_serial** statement, defining a **%stop_try** clause prevents further execution of the **ksh** statement on all cluster nodes; it also prevents further error checking defined by the **%except** clause.

In a **%try_parallel** statement, defining a **%stop_try** clause prevents error checking on all cluster nodes, since execution of the **try** statement happens simultaneously on all nodes. The commands within the **ksh** statement will have completed on other nodes by the time the **%except** clauses are evaluated.

%others clause:

```
%others error_id
       ksh code
       [ %stop_try clause ]
       ksh code
[ %end | %end_o ]
```

The **%others** clause is the default error action performed if a command's return code does *not* match the return code of a specific **%except** clause. The **%others** clause can be used in a **try** statement, or globally within an execution plan.

%end statement:

```
%end
```

The **%end** statement is used with all CEL constructs. Ensure that your **.cel** file includes an **%end** statement with each statement or clause. A construct like **%try_parallel**, for example, can be ended with **%end** or with **%end_p**, where **p** represents parallel.

Actual Execution Plan

The following example shows an actual execution plan, **cl_chuser.cel**, written in CEL that uses the required constructs to create the **cl_chuser** command. The CEL constructs are shown in bold. Other examples of execution plans are in the **/usr/es/sbin/cluster/samples/cspoc** directory.

```
#####################################################################
# Name:
#       cl_chuser.cel
#
# Description:
#   The cl_chuser command changes user attributes for a user on all  # nodes in an
HACMP cluster.
#
#   Usage: cl_chuser [-cspoc "[-f]"] Attribute=Value ... Name
#
# Arguments:
#   The cl_chuser command arguments include all options and
#   arguments that are valid for the chuser command as well as
#   C-SPOC specific arguments. The C-SPOC specific arguments are as #follows:
#      -f                C-SPOC force flag
#
# Return Values:
#       0       success
#       1       failure
#
#####################################################################
# Initialize variables
_CMD_NAME=`basename $0`
_CSPOC_OPT_STR="d:f"
_OPT_STR=""
_USAGE="Usage: cl_chuser [-cspoc #7f>[-f]] Attribute=Value ... Name"
_MSET=12
```

```
_RETCODE=0
# Default exception handling for a COULD_NOT_CONNECT error
%except 1000
        nls_msg -l $cspoc_tmp_log ${_MSET} CDSH_UNABLE_TO_CONNECT "${_CMD_NAME}:
Unable to
 connect to node ${_NODE}" >& 2
        if [[ -z "${_SPOC_FORCE}" ]]
        then
                exit 1
        fi
%end
# C-SPOC Initialization and Verification
%include cl_path.cel
%include cl_init.cel
%define USER_NOT_FOUND 2
user=${_CMD_ARGS##* }
if [[ -z "${_SPOC_FORCE}" ]]
then
  #
  # Check if user exists across all cluster nodes
  #
  %try_parallel _NODE _TARGET_NODES silent_err silent_out
          lsuser $user
          %except USER_NOT_FOUND
          nls_msg -l $cspoc_tmp_log ${_MSET} 1 "${_CMD_NAME}: User ${user} does not
exist on node ${_NODE}" ${_CMD_NAME} ${user} ${_NODE} >& 2
                  _RETCODE=1
        %end
  %end
fi
# If user does not exist on a node, exit 1
if [[ ${_RETCODE} -ne 0 ]]
then
  exit ${_RETCODE}
fi
# Run chuser across the cluster nodes
%try_parallel _NODE _TARGET_NODES
  chuser $_CMD_ARGS
  %others
          # If chuser returned an error on any node, exit 1
          _RETCODE=1
  %end
%end
exit ${_RETCODE}
```

Converting Execution Plans to ksh Scripts

After writing an execution plan for a command, use the CEL preprocessor (**celpp**) to convert it into a **ksh** script; then execute the **chmod** command on the script to make it executable. An executable **ksh** script includes all code required to implement the command across cluster nodes. The preprocessor generates much of the code you would otherwise have to write.

To convert an execution plan into a **ksh** script:

1. Change directories to the one containing the execution plan (**.cel** file) for the command you want to convert.
2. Enter the following command to run **celpp**, specifying the **.cel** file as the input file and the command to be generated as the output file:

   ```
   celpp [-i inputfile] [-o outputfile] [-I IncludeDirectory]
   ```

where the elements of this command line are:

- **-i** *inputfile* Uses *inputfile* for input, or **stdin** by default.
- **-o** *outputfile* Uses *outputfile* for output, or **stdout** by default.
- **-I** *IncludeDirectory* Uses the specified directory name to locate **.cel** files (execution plans). You can specify multiple **-I** options. The **cl_init.cel** and **cl_path.cel** files are installed in **/usr/es/sbin/cluster/samples/cspoc**; The IncludeDirectory should normally be specified.

For example, enter:

```
celpp -i cl_chuser.cel -o cl_chuser -I
/usr/es/sbin/cluster/samples/cspoc]
```

This command converts the **cl_chuser.cel** file into a **ksh** script.

3. Enter the following command to make the generated **ksh** script executable:

   ```
   chmod +x inputfile
   ```

   For example, enter:

   ```
   chmod +x cl_chuser
   ```

You can now invoke the C-SPOC command. If you change a command's execution plan after converting it, repeat the preceding steps to generate a new **ksh** script. Then make the script executable. Note that the preceding steps can be included in a **Makefile**.

Storing C-SPOC Commands

The HACMP for AIX software must be installed in your cluster to use C-SPOC commands. You can store C-SPOC commands you create in any directory.

Handling Command Output

When you are ready to use a C-SPOC command, be aware that all error output (messages) resulting from the command's execution across the cluster is returned to **stdout** and **stderr**. By default, **stdout** and **stderr** remain directed to your display. You can override this action by using the **silent_err** or **silent_out** arguments with the **try** statements in a command's execution plan. If the **silent_err** flag is defined, **stderr** from the enclosed **try ksh** statement will appear only in the **$try_err** file. If the **silent_out** flag is defined, **stdout** from the enclosed **try ksh** statement will appear only in the **$try_out** file. Refer to Writing an Execution Plan for an example of how you use these arguments.

Whatever way you direct error output, all error messages are prefixed with the name of the node on which the command was executed. When the command completes on each node, error messages are logged in the **cspoc.log** file—the log file for all C-SPOC commands. for more information about C-SPOC-related log entries, see Chapter 2: Using Cluster Log Files.

Appendix C: HACMP Tracing

This appendix describes how to trace HACMP-related events.

HACMP Tracing Overview

The trace facility helps you isolate a problem within an HACMP system by allowing you to monitor selected events. Using the trace facility, you can capture a sequential flow of time-stamped system events that provide a fine level of detail on the activity within an HACMP cluster.

The trace facility is a low-level debugging tool that augments the troubleshooting facilities described earlier in this book. While tracing is extremely useful for problem determination and analysis, interpreting a trace report typically requires IBM support.

The trace facility generates large amounts of data. The most practical way to use the trace facility is for short periods of time—from a few seconds to a few minutes. This should be ample time to gather sufficient information about the event you are tracking and to limit use of space on your storage device.

The trace facility has a negligible impact on system performance because of its efficiency.

Using the Trace Facility for HACMP Daemons

Use the trace facility to track the operation of the following HACMP daemons:

- The Cluster Manager daemon (**clstrmgrES**)
- The Cluster Information Program daemon (**clinfoES)**
- The Cluster Communication daemon (**clcomdES**)

The **clstrmgrES**, **clinfoES** and **clcomd** daemons are controlled by the System Resource Controller (SRC).

Enabling Daemons under the Control of the System Resource Controller

The **clstrmgrES**, and **clinfoES** daemons are user-level applications under the control of the SRC. Before you can start a trace on one of these daemons, enable tracing for that daemon. *Enabling* tracing on a daemon adds that daemon to the master list of daemons for which you want to record trace data.

clcomd Daemon

The **clcomd** daemon is also under control of the SRC. To start a trace of this daemon, use the AIX **traceson** command and specify the **clcomd** subsystem.

Initiating a Trace Session

Use SMIT to initiate a trace session for the **clstrmgr** or **clinfo** utilities. SMIT lets you enable tracing in the HACMP SRC-controlled daemons, start and stop a trace session in the daemons, and generate a trace report. The following sections describe how to use SMIT for initiating a trace session.

Using SMIT to Obtain Trace Information

To initiate a trace session using the SMIT interface:

1. Enable tracing on the SRC-controlled daemon or daemons you specify.

   Use the SMIT **Problem Determination Tools** > **HACMP Trace Facility > Enable/Disable Tracing of HACMP for AIX Daemons** panel to indicate that the selected daemons should have trace data recorded for them.

2. Start the trace session.

   Use the SMIT **Start/Stop/Report Tracing of HACMP for AIX Services** panel to trigger the collection of data.

3. Stop the trace session.

   You must stop the trace session before you can generate a report. The tracing session stops either when either you use the SMIT **Start/Stop/Report Tracing of HACMP for AIX Services** panel to stop the tracing session or when the log file becomes full.

4. Generate a trace report.

   Once the trace session is stopped, use the SMIT **Start/Stop/Report Tracing of HACMP for AIX Services** panel to generate a report.

Each step is described in the following sections.

Enabling Tracing on SRC-controlled Daemons

To enable tracing on the following SRC-controlled daemons (**clstrmgrES or clinfoES**):

1. Enter: `smit hacmp`
2. Select **Problem Determination Tools** > **HACMP Trace Facility** and press Enter.
3. Select **Enable/Disable Tracing of HACMP for AIX Daemons** and press Enter.
4. Select **Start Trace** and press Enter. SMIT displays the Start Trace panel. Note that you only use this panel to *enable* tracing, *not* to actually start a trace session. It indicates that you want events related to this particular daemon captured the next time you start a trace session. See Starting a Trace Session for more information.
5. Enter the PID of the daemon whose trace data you want to capture in the **Subsystem PROCESS ID** field. Press F4 to see a list of all processes and their PIDs. Select the daemon and press Enter. Note that you can select *only* one daemon at a time. Repeat these steps for each additional daemon that you want to trace.

6. Indicate whether you want a short or long trace event in the **Trace Type** field. A short trace contains terse information. For the **clstrmgrES** daemon, a short trace produces messages only when topology events occur. A long trace contains detailed information on time-stamped events.
7. Press Enter to enable the trace. SMIT displays a panel that indicates that tracing for the specified process is enabled.

Disabling Tracing on SRC-controlled Daemons

To disable tracing on the **clstrmgrES or clinfoES** daemons:

1. Enter: `smit hacmp`
2. Select **Problem Determination Tools > HACMP Trace Facility > Enable/Disable Tracing of HACMP for AIX Daemons > Stop Trace**. SMIT displays the **Stop Trace** panel. Note that you only use this panel to *disable* tracing, *not* to actually stop a trace session. It indicates that you do *not* want events related to this particular daemon captured the next time you run a trace session.
3. Enter the PID of the process for which you want to disable tracing in the **Subsystem PROCESS ID** field. Press F4 to see a list of all processes and their PIDs. Select the process for which you want to disable tracing and press Enter. Note that you can disable only one daemon at a time. To disable more than one daemon, repeat these steps.
4. Press Enter to disable the trace. SMIT displays a panel that indicates that tracing for the specified daemon has been disabled.

Starting a Trace Session

Starting a trace session triggers the actual recording of data on system events into the system trace log from which you can later generate a report.

Remember, you can start a trace on the **clstrmgrES** and **clinfoES** daemons only if you have previously enabled tracing for them.

To start a trace session:

1. Enter: `smit hacmp`
2. Select **Problem Determination Tools > HACMP Trace Facility > Start/Stop/Report Tracing of HACMP for AIX Services > Start Trace**. SMIT displays the **Start Trace** panel.
3. Enter the trace IDs of the daemons that you want to trace in the **ADDITIONAL event IDs to trace** field.

   Press F4 to see a list of the trace IDs. (Press Ctrl-v to scroll through the list.) Move the cursor to the first daemon whose events you want to trace and press F7 to select it. Repeat this process for each event that you want to trace. When you are done, press Enter. The values that you selected are displayed in the **ADDITIONAL event IDs to trace** field. The HACMP daemons have the following trace IDs:

clstrmgrES	910
clinfoES	911

4. Enter values as necessary into the remaining fields and press Enter. SMIT displays a panel that indicates that the trace session has started.

Stopping a Trace Session

You need to stop a trace session before you can generate a trace report. A trace session ends when you actively stop it or when the log file is full.

To stop a trace session.

1. Enter `smit hacmp`
2. In SMIT, select **Problem Determination Tools** > **HACMP Trace Facility > Start/Stop/Report Tracing of HACMP for AIX Services > Stop Trace**. SMIT displays the Command Status panel, indicating that the trace session has stopped.

Generating a Trace Report

A trace report formats the information stored in the trace log file and displays it in a readable form. The report displays text and data for each event according to the rules provided in the trace format file.

When you generate a report, you can specify:

- Events to include (or omit)
- The format of the report.

To generate a trace report:

1. Enter: `smit hacmp`
2. In SMIT, select **Problem Determination Tools > HACMP Trace Facility > Start/Stop/Report Tracing of HACMP for AIX Services > Generate a Trace Report**. A dialog box prompts you for a destination, either a filename or a printer.
3. Indicate the destination and press Enter. SMIT displays the Generate a Trace Report panel.
4. Enter the trace IDs of the daemons whose events you want to include in the report in the **IDs of events to INCLUDE in Report** field.
5. Press F4 to see a list of the trace IDs. (Press Ctrl-v to scroll through the list.) Move the cursor to the first daemon whose events you want to include in the report and press F7 to select it. Repeat this procedure for each event that you want to include in the report. When you are done, press Enter. The values that you selected are displayed in the **IDs of events to INCLUDE in Report** field.The HACMP daemons have the following trace IDs:

clstrmgrES	910
clinfoES	911

6. Enter values as necessary in the remaining fields and press Enter.
7. When the information is complete, press Enter to generate the report. The output is sent to the specified destination. For an example of a trace report, see the following Sample Trace Report section.

Sample Trace Report

The following is a sample trace report.

```
Wed Mar  10 13:01:37 1998
System: AIX steamer Node: 3
Machine: 000040542000
Internet Address: 00000000 0.0.0.0

trace -j 011  -s -a

ID  PROCESS NAME   I SYSTEM CALL   ELAPSED     APPL    SYSCALL KERNEL  INTERRUPT

001 trace                          0.000000    TRACE ON channel 0
Fri Mar 10 13:01:38 1995
011 trace                         19.569326   HACMP for AIX:clinfo Exiting Function:
broadcast_map_request
011 trace                          19.569336   HACMP for AIX:clinfo Entering
Function: skew_delay
011 trace                         19.569351   HACMP for AIX:clinfo Exiting Function:
skew_delay, amount: 718650720
011 trace                         19.569360   HACMP for AIX:clinfo Exiting Function:
service_context
011 trace                          19.569368   HACMP for AIX:clinfo Entering
Function: dump_valid_nodes
011 trace                          19.569380   HACMP for AIX:clinfo Entering
Function: dump_valid_nodes
011 trace                          19.569387   HACMP for AIX:clinfo Entering
Function: dump_valid_nodes
011 trace                          19.569394   HACMP for AIX:clinfo Entering
Function: dump_valid_nodes
011 trace                          19.569402   HACMP for AIX:clinfo Waiting for event
011 trace                          22.569933   HACMP for AIX:clinfo Entering
Function: service_context
011 trace                          22.569995   HACMP for AIX:clinfo Cluster ID: -1
011 trace                          22.570075   HACMP for AIX:clinfo Cluster ID: -1
011 trace                          22.570087   HACMP for AIX:clinfo Cluster ID: -1
011 trace                          22.570097   HACMP for AIX:clinfo Time Expired: -1
011 trace                          22.570106   HACMP for AIX:clinfo Entering
Function: broadcast_map_request
002 trace                          23.575955              TRACE OFF channel 0
                                                          Wed Nov 15 13:02:01 1999
```

Notices for HACMP Troubleshooting Guide

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 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 country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Dept. LRAS / Bldg. 003
11400 Burnet Road
Austin, TX 78758-3493
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.

Index

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W