PRIMECLUSTER™

Reliant Monitor Services (RMS) (Solaris®, Linux®)
Troubleshooting Guide

Edition November 2004

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion of this manual. Your feedback helps us optimize our documentation to suit your individual needs.

Fax forms for sending us your comments are included in the back of the manual.

There you will also find the addresses of the relevant User Documentation Department.

Certified documentation according DIN EN ISO 9001:2000

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2000.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright and Trademarks

This manual is printed on
paper treated with
chlorine-free bleach.

Glossary

Abbreviations

Figures

Tables

Index

Contents

Contents

Contents

1 Preface

Reliant® Monitor Services (RMS) is a software monitor designed to guarantee the high availability of applications in a cluster of nodes. This manual describes how to troubleshoot RMS configuration and operation.

The manual is aimed at RMS experts who create and tune RMS configurations. They should be familiar with the RMS Wizards Tools or PCS, the RMS CLI commands and utilities, and the following system functions and components:

● PRIMECLUSTER family of products

● Solaris® or Linux® operating system

● Non-PRIMECLUSTER products such as volume managers and storage area networks.

1.1 About this manual

This manual is structured as follows:

● The chapter "Troubleshooting" on page 9 describes how to troubleshoot RMS using graphical user interface (GUI) and command line interface (CLI) tools.

● The chapter "Non-fatal error messages" on page 39 lists all RMS error messages written to the log file along with their causes and resolutions.

● The chapter "Fatal error messages" on page 125 lists all fatal RMS error messages written to the log file along with their causes and resolutions.

● The chapter "Console error messages" on page 141 lists all RMS error messages written to the console along with their causes and resolutions.

● The chapter "Appendix—Operating system error numbers" on page 167 lists operating system error numbers for Solaris and Linux.

● The chapter "Appendix—Object types" on page 179 lists the object types that are supplied with RMS.

● The chapter "Appendix—Attributes" on page 181 lists the attributes that are supported by RMS object types.

● The chapter "Appendix—Environment variables" on page 195 describes the RMS environment variables.

● The chapter "Appendix—List of manual pages" on page 207 lists the manual pages for PRIMECLUSTER.

1.2 PRIMECLUSTER documentation list

The documents listed below provide details about PRIMECLUSTER products. Please contact your sales representative for ordering information.

● *Concepts Guide (Solaris, Linux)*—Provides conceptual details on the PRIME-CLUSTER family of products.

● *Adaptive Services (AS) (Linux, Windows) User Guide*—Describes how to install, configure, and administer Adaptive Services core and satellite nodes.

● *PCS for Adaptive Services (AS) (Linux) Configuration Guide*—Describes how to use PCS (PRIMECLUSTER Configuration Services) to configure application monitoring for Adaptive Services.

● *Installation Guide (Solaris)*—Provides instructions for installing and upgrading PRIMECLUSTER products.

● *Installation Guide (Linux)*—Provides instructions for installing and upgrading PRIMECLUSTER products.

● *PRIMECLUSTER for mySAP Business Suite Installation Guide (Linux)*—Provides instructions for installing PRIMECLUSTER for mySAP Business Suite on Linux.

● *Web-Based Admin View (Solaris) Operation Guide*—Provides information on using the Web-Based Admin View management GUI.

● *Web-Based Admin View (Linux) Operation Guide*—Provides information on using the Web-Based Admin View management GUI.

● *Cluster Foundation (CF) (Solaris) Configuration and Administration Guide*—Provides instructions for configuring and administering the PRIME-CLUSTER Cluster Foundation.

● *Cluster Foundation (CF) Configuration and Administration Guide (Linux)*—Provides instructions for configuring and administering the PRIME-CLUSTER Cluster Foundation.

● *PRIMECLUSTER for mySAP Business Suite Shutdown Facility (SF) (Linux) Configuration and Administration Guide*—Provides instructions for configuring, administering and troubleshooting the PRIMECLUSTER for mySAP Business Suite Shutdown Facility.

- *PRIMECLUSTER for mySAP Business Suite Reliant Monitor Services (RMS) with Wizard Tools (Linux) Configuration and Administration Guide*—Provides instructions for configuring and administering PRIMECLUSTER for mySAP Business Suite Reliant Monitor Services using the Wizard Tools interface

- *PRIMECLUSTER for mySAP Business Suite Reliant Monitor Services (RMS) with PCS (Linux) Configuration and Administration Guide*—Provides instructions for configuring and administering PRIMECLUSTER for mySAP Business Suite Reliant Monitor Services using the PCS (PRIMECLUSTER Configuration Services) interface

- *Reliant Monitor Services (RMS) with Wizard Tools (Solaris, Linux) Configuration and Administration Guide*—Provides instructions for configuring and administering PRIMECLUSTER Reliant Monitor Services using the Wizard Tools interface.

- *Reliant Monitor Services (RMS) with PCS (Solaris, Linux) Configuration and Administration Guide*—Provides instructions for configuring and administering PRIMECLUSTER Reliant Monitor Services using the PCS (PRIME-CLUSTER Configuration Services) interface.

- *Reliant Monitor Services (RMS) (Solaris, Linux) Troubleshooting Guide*—Describes diagnostic procedures to solve RMS configuration problems, including how to view and interpret RMS log files. Provides a list of all RMS error messages with a probable cause and suggested action for each condition.

- *Scalable Internet Services (SIS) (Solaris, Linux) Configuration and Administration Guide*—Provides information on configuring and administering Scalable Internet Services (SIS).

- *Global Disk Services (Solaris) Configuration and Administration Guide*—Provides information on configuring and administering Global Disk Services (GDS).

- *Global File Services (Solaris) Configuration and Administration Guide*—Provides information on configuring and administering Global File Services (GFS).

- *Global Link Services (Solaris) Configuration and Administration Guide: Redundant Line Control Function*—Provides information on configuring and administering the redundant line control function for Global Link Services (GLS).

- *Global Link Services (Solaris) Configuration and Administration Guide: Multipath Function*—Provides information on configuring and administering the multipath function for Global Link Services (GLS).

- *Data Management Tools (Solaris) Configuration and Administration Guide*—Provides reference information on the Volume Manager (RCVM) and File Share (RCFS) products. (Not available in all markets)

- *SNMP Reference Manual (Solaris, Linux)*—Provides reference information on the Simple Network Management Protocol (SNMP) product.

- Release notices for all products—These documentation files are included as HTML files on the PRIMECLUSTER Framework CD. Release notices provide late-breaking information about installation, configuration, and operations for PRIMECLUSTER. Read this information first.

- *RMS Wizards documentation package*—Available on the PRIMECLUSTER CD. These documents deal with Wizard Tools topics such as the configuration of file systems and IP addresses. They also describe the various types of available RMS wizards.

- PCS Migration Guide—Available on the PRIMECLUSTER CD. Describes procedures for converting configurations from the Wizard Tools environment to PCS.

1.3   Conventions

To standardize the presentation of material, this manual uses a number of notational, typographical, and syntactical conventions.

1.3.1   Notation

This manual uses the following notational conventions.

1.3.1.1   Prompts

Command line examples that require system administrator (or root) rights to execute are preceded by the system administrator prompt, the hash sign (#). Entries that do not require system administrator rights are preceded by a dollar sign ($).

In some examples, the notation *<nodename>*# indicates a root prompt on the specified node. For example, a command preceded by `fuji2#` would mean that the command was run as user `root` on the node named `fuji2`.

1.3.1.2   Manual page section numbers

References to operating system commands are followed by their manual page section numbers in parentheses—for example, `cp`(1).

1.3.1.3   The keyboard

Keystrokes that represent nonprintable characters are displayed as key icons such as ⎡Enter⎤ or ⎡F1⎤. For example, ⎡Enter⎤ means press the key labeled *Enter*; ⎡Ctrl-b⎤ means hold down the key labeled *Ctrl* or *Control* and then press the ⎡B⎤ key.

1.3.1.4   Typefaces

The following typefaces highlight specific elements in this manual.

Typeface	Usage
Constant Width	Computer output and program listings; commands, file names, manual page names and other literal programming elements in the main body of text.
Italic	Variables in a command line that you must replace with an actual value. May be enclosed in angle brackets to emphasize the difference from adjacent text, e.g., *<nodename>*RMS; unless directed otherwise, you should not enter the angle brackets. The name of an item in a character-based or graphical user interface. This may refer to a menu item, a radio button, a checkbox, a text input box, a panel, or a window title.
Bold	Items in a command line that you must type exactly as shown.

Typeface conventions are shown in the following examples.

1.3.1.5   Example 1

Several entries from an /etc/passwd file are shown below:

```
root:x:0:1:0000-Admin(0000):/:/sbin/ksh
sysadm:x:0:0:System Admin.:/usr/admin:/usr/sbin/sysadm
setup:x:0:0:System Setup:/usr/admin:/usr/sbin/setup
daemon:x:1:1:0000-Admin(0000):/:
```

1.3.1.6   Example 2

To use the cat(1) command to display the contents of a file, enter the following command line:

$ cat *file*

1.3.2   Command syntax

The command syntax observes the following conventions.

Symbol	Name	Meaning
[ ]	Brackets	Enclose an optional item.
{ }	Braces	Enclose two or more items of which only one is used. The items are separated from each other by a vertical bar (│).
│	Vertical bar	When enclosed in braces, it separates items of which only one is used. When not enclosed in braces, it is a literal element indicating that the output of one program is piped to the input of another.
( )	Parentheses	Enclose items that must be grouped together when repeated.
...	Ellipsis	Signifies an item that may be repeated. If a group of items can be repeated, the group is enclosed in parentheses.

1.4 Important notes and cautions

Material of particular interest is preceded by the following symbols in this manual:

i   Contains important information about the subject at hand.

Caution

Indicates a situation that can cause harm to data.

2 Troubleshooting

This chapter discusses some PRIMECLUSTER facilities for debugging the RMS product from both the command line interface (CLI) and from the Cluster Admin graphical user interface (GUI). This chapter provides details on log files, their location, how to turn on logging levels, how to view logs from the GUI, and how to view log files from CLI.

This chapter discusses the following:

● The section "Overview" on page 9 summarizes the troubleshooting process.

● The section "Debug and error messages" on page 11 describes RMS debug and error messages.

● The section "Log files" on page 12 identifies and explains the RMS log files.

● The section "Using the log viewer" on page 14 explains the log viewer facilities.

● The section "Specifying the log level" on page 22 specifies and explains the log levels.

● The section "Interpreting log files" on page 24 explains the meaning of the data in the log files.

● The section "System log" on page 25 describes the system log.

● The section "PCS and Wizard Tools log files" on page 26 details the RMS Wizard log files.

● The section "PCS troubleshooting" on page 32 lists the locations of the PCS log files.

● The section "RMS troubleshooting" on page 33 supplies solutions to problems that could occur while using RMS.

2.1 Overview

The RMS troubleshooting process usually begins after you observe an error condition or state change in Cluster Admin in one of the following areas:

● Clusterwide table

● RMS tree

● Graph

The clusterwide table contains summary information and is a good place to start looking for error conditions. For additional details, you can look at the RMS tree or the graph. Depending on whether you need to look at the switchlogs or application logs, you can then use the log viewer facility to view the log files.

The log viewer has search facilities based on the following:

● Keywords

● Severity

● Non-zero exit codes

Search for causes of errors using the keywords and the date range fields. For emergency, alert, and critical conditions, you can do a search based on severity. For proactive troubleshooting, you can perform a search based on severity for the `error`, `warning`, `notice`, and `info` severity codes.

> **i** It is recommended that you periodically use the log viewer and check the log files based on the severity levels to avoid serious problems. If you cannot diagnose the cause of a problem, look at the log viewer from two or more nodes in the cluster.

> **i** Refer to the section "RMS troubleshooting" on page 33 for an explanation on corrective action.

Resolve error conditions as follows:

1. Use the Cluster Admin GUI.

2. View the log files if needed.

3. Change log levels to get more details.

4. If you cannot resolve an error condition with the GUI, you can use the command line interface. Use standard UNIX commands.

5. If a problem persists, check if it is a non-RMS issue and refer to the appropriate manual.

6. Check for system-related issues like operating system, hardware, or network errors.

7. Contact field support if you cannot resolve the issue.

2.2   Debug and error messages

RMS writes debug and error messages to log files when its components (such as the base monitor or detectors) operate. The default setting is for RMS to store these files in the `/var/opt/SMAWRrms/log` directory. Users can change the directory with the `RELIANT_LOG_PATH` environment variable, which is set in the `hvenv.local` file.

When RMS starts, logging begins. The default setting is for the base monitor to write all error messages to its log file or to `stderr`. Normally, you do not need to change the default setting because the default options allow for very detailed control of debug output.

If required, you can use the base monitor to record every state and message of any node. However, in most cases, the information requires a detailed knowledge of internal RMS operation to interpret the debug output, which can only be evaluated by service personnel.

For the administrator of an RMS cluster, evaluating the `switchlog` file is normally sufficient. This file records all important RMS actions; for example, incoming switch requests or faults that occur in nodes or resources.

| i | There are also configuration-specific log files in the log directory. It is recommended that administrators evaluate these if necessary. The name of each log file (normally *<userApplication>*`.log`) depends on the configuration that was set up using the configuration wizards (RMS Wizard Tools or PCS). Consult the RMS Wizard Tools or PCS online documentation for further information. |

The `bmlog` log file can also be useful for problem solving.

2.3   Log files

Table 1 identifies and explains the RMS log files contained in
`/var/opt/SMAWRrms/log`.

Module	File Name	Contents
base monitor	`tracelog`	Records all messages between objects and all modification instructions. By default, RMS places no messages in `tracelog`.
base monitor	`abortstartlog`	This file contains records about bm exit conditions to assist support personnel in determining why RMS failed to start. This file is generated if the following message appears during startup: `FATAL ERROR: RMS has failed to start!`

Table 1: Log files

Module	File Name	Contents
base monitor	bmlog	General RMS error and message logging information ranges from simple message reporting to more complete information. The error log level determines the contents of this file, which is specified when the base monitor is started. Refer to the section "Specifying the log level" on page 22 for more information.
		Includes all messages received by the base monitor at runtime.
		Should be used by experts only, since turning on log level flags consumes a great deal of disk space. By default, RMS places no messages in bmlog.
Everything (base monitor, generic detector, node detector)	switchlog	Operational events, such as resource switches or bugs. Normally, switchlog is the only log file users need to examine.
generic detector	*detname*.log	All messages and job assignments received by the detector. Also contains resource state change information and all error messages.
		detname is the name of the detector (hvdet_* for RMS Wizard Tools and pcsdet_* for PCS) in the *<RELIANT_PATH>*/bin directory.

Table 1: Log files

2.4   Using the log viewer

Using the Cluster Admin interface, you can view two types of RMS log messages on each node: the RMS switchlog, and individual application logs.

View the switchlog for a system node as follows:

▶ Right-click on the system node and select *View Switchlog* from the pop-up context menu (Figure 1). Alternatively, select a node and use *Tools –> View switchlog* (Figure 2).
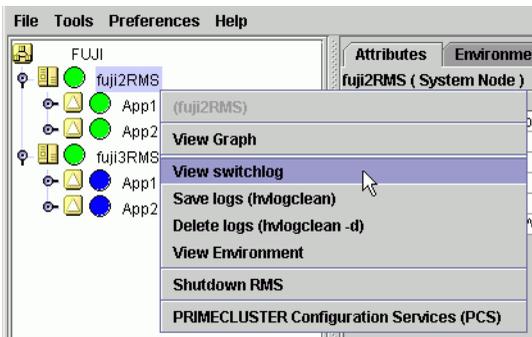
Figure 1: Viewing the RMS switchlog file using a context menu
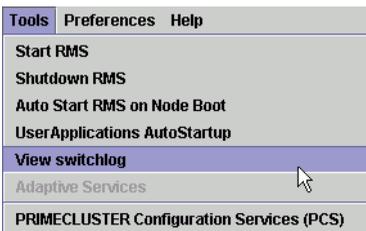
Figure 2: Viewing the RMS switchlog file using the Tools menu

View an application log as follows:

▶ Right-click on an application on the RMS tree and choose *View logfile* from the pop-up context menu (Figure 3).
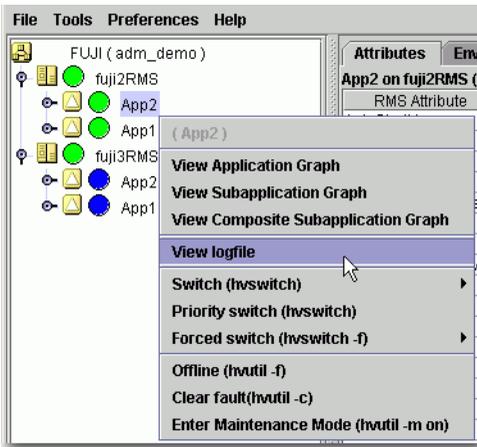
Figure 3: Viewing an application log using a context menu

Each log file is displayed in a separate tab in the right pane (Figure 4).
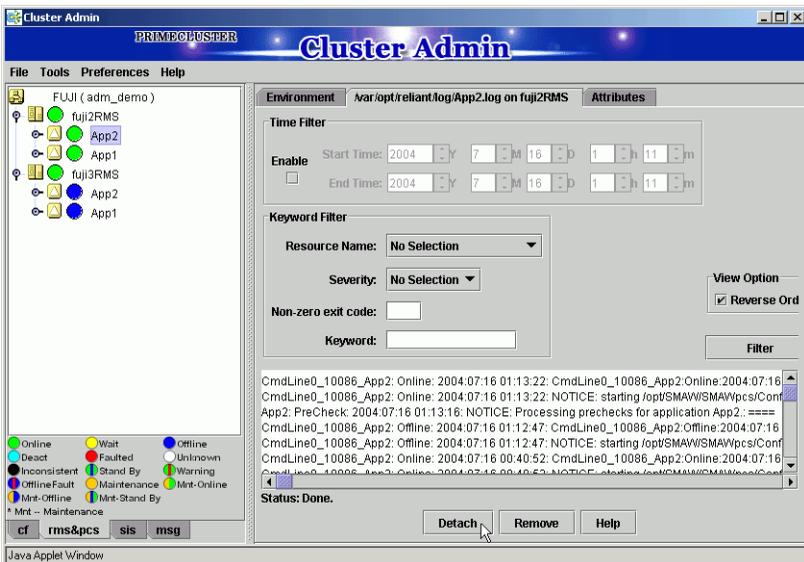
Figure 4: Viewing an RMS log

The *Detach* button at the bottom of the view will separate the current log display tab and display it in its own window (Figure 5). The detached window can be rejoined to the main window with the *Attach* button.
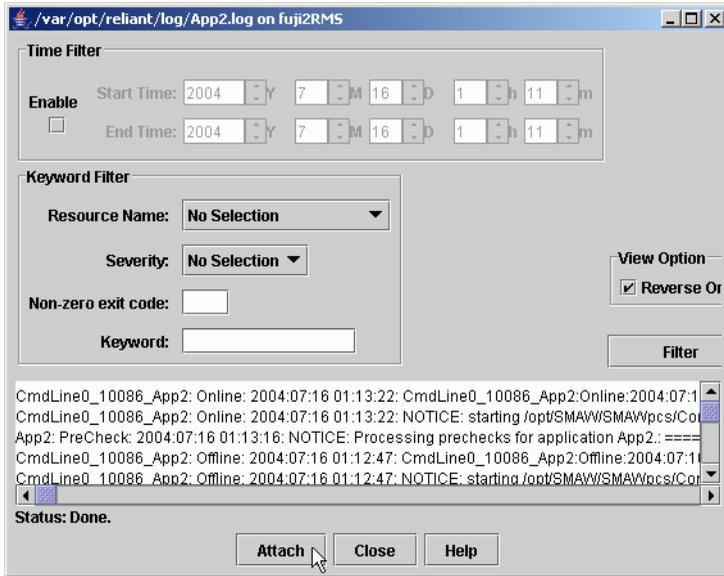
Figure 5: Viewing the RMS switchlog file in a detached window

> **i** All RMS log files, which normally reside in `/var/opt/SMAWRrms/log/`, can be viewed directly using a standard UNIX editor like vi.

Common procedures for switchlog and application log viewing

By default, the entire log is available in the scrolled area at the bottom of the window. You can restrict the entries displayed with the following filters:

● Timestamp: Click the *Enable* check box and select the period of interest.

● Resource name (for an application only), severity of error messages, non-zero exit code, or keyword. Selected and non-blank criteria are combined with a logical *and*.

> **i** Refer to the *RMS Troubleshooting Guide* for a complete description of severity levels and exit codes.

After you enter your filter criteria, click the *Filter* button to display the filtered log entries.

At any time, you can sort the displayed switchlog entries according to increasing or decreasing time by checking or unchecking the *View in Reverse Order* checkbox in the log viewer window.

Time filter

Figure 6 shows the view for a search based on the date and time.
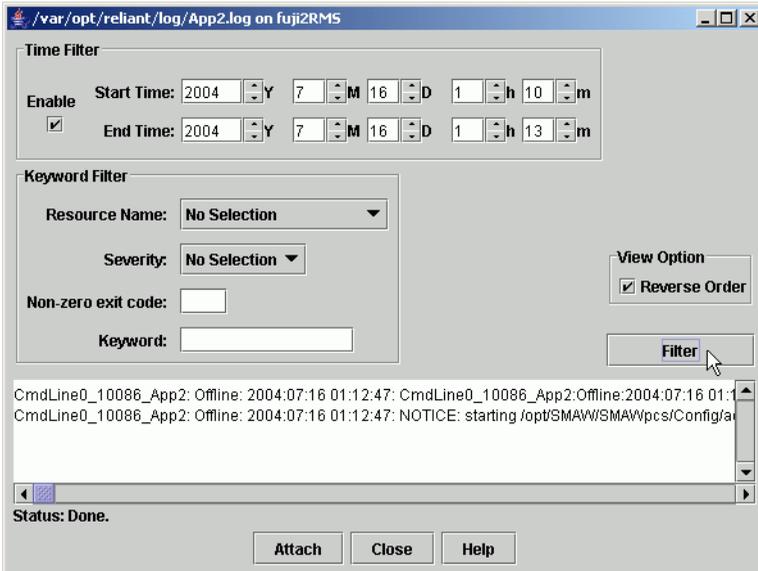
Figure 6: Search based on date and time range

Resource name

Figure 7 shows the view for a search based on a resource name. This search is enabled only in an application log.
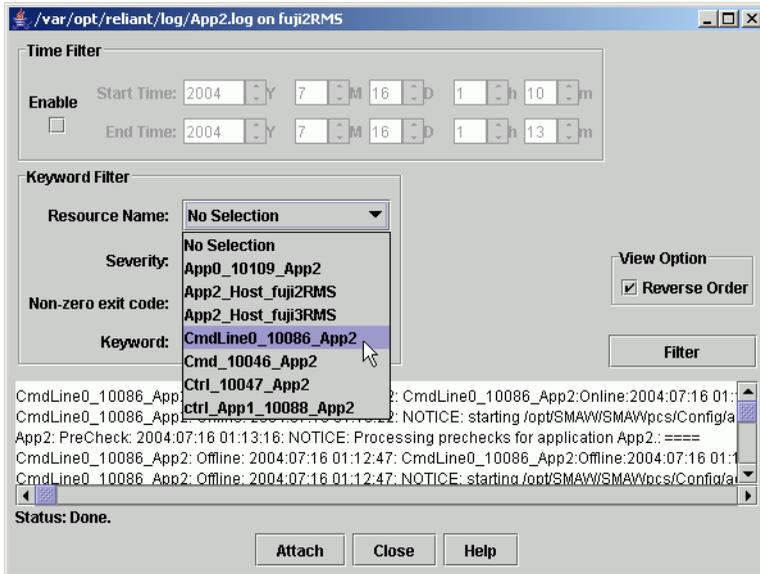
Figure 7: Search based on resource name

Severity

Figure 8 shows the view for a search based on the severity level.
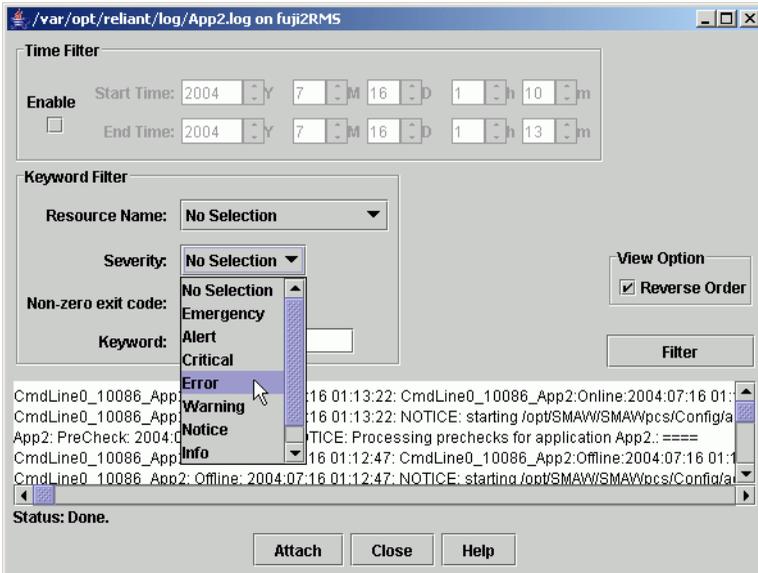
Figure 8: Search based on severity level

Table 2 summarizes the RMS severity levels.

Severity level	Description
Emergency	Systems cannot be used
Alert	Immediate action is necessary
Critical	Critical condition
Error	Error condition
Warning	Warning condition
Notice	Normal but important condition
Info	Miscellaneous information
Debug	Debug messages

Table 2: RMS severity level description

Exit code

Enter a numeric exit code in the *Non zero exit code box*.

Keyword

Enter a string in the *Keyword* box. Special characters and spaces are valid, but wildcards are not interpreted. The search is case-insensitive. Figure 9 shows the view for a search based on a string.
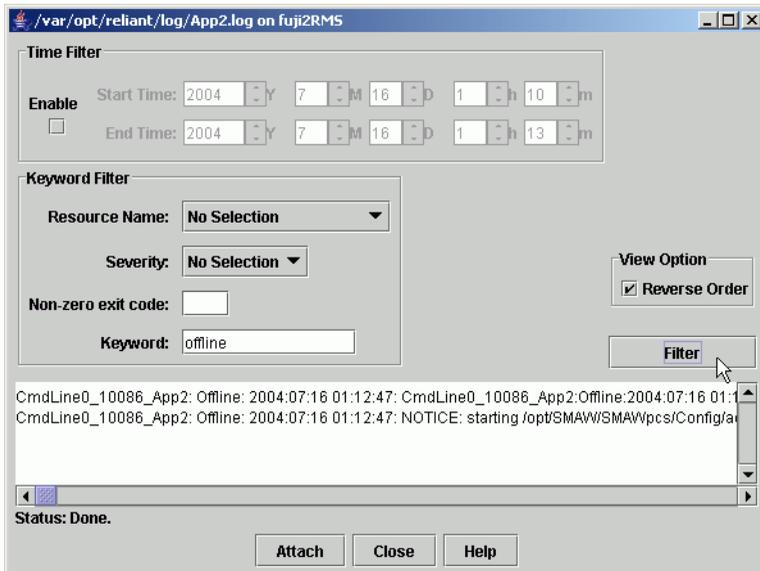
Figure 9: Search based on keyword

You can also search the text in the application log by right-clicking on the displayed text. This brings up a small command pop-up with a *Find* entry that allows you to perform a case-sensitive search (Figure 10).
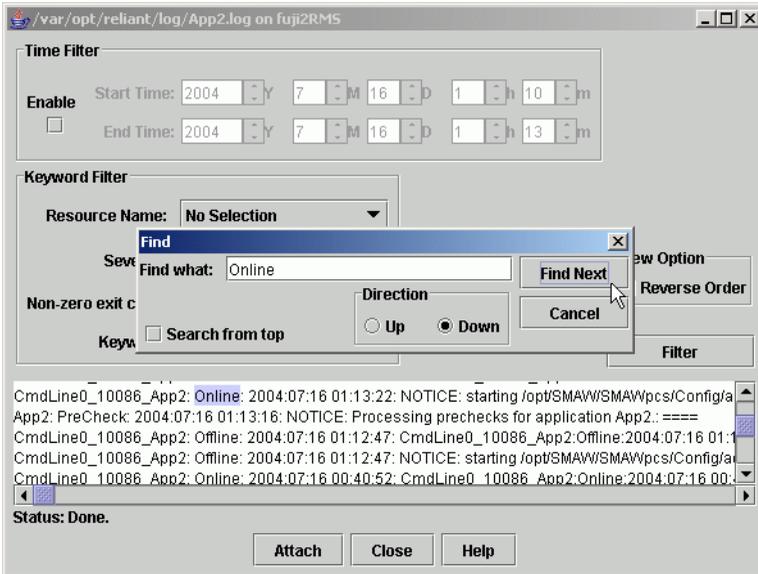
Figure 10: Using the Find pop-up in log viewer

Removing filters

To remove all filters, take the following steps:

● Uncheck the time filter *Enable* box.

● Set drop-down lists to *No Selection*

● Clear text from input boxes

● Click the *Filter* button

The unfiltered view will be restored.

2.5   Using the hvdump command

The `hvdump` command is used to get debugging information about RMS on the local node. Independent of the base monitor running on the local node, invoking `hvdump` causes it to gather PRIMECLUSTER product and system files that will be used to diagnose the problem. For a detailed list of the information gathered, consult the `hvdump`(1M) manual page.

2.6   Specifying the log level

For further debugging information, use the '−l *level*' option of the `hvcm` or `hvutil` commands to activate various logging procedures.

i Specify logging with the −l (lowercase "L") flag.

To activate logging, start RMS with one of the log levels described in Table 3, or use the `hvutil` command to set the logging after RMS has been started. The log level specified with the −l option is a list with numbers or a range. Separate levels by means of commas or spaces in the list. If a space is used as a list separator, include the entire argument between the braces. A level range is defined as *n1−n2*. This includes all log levels from *n1* up to and including *n2*. Specifying *−n2* is the same as 1−*n2*. Specifying *n1−* includes all log levels above *n1*. The *n1* value must be greater than or equal to 1.

All log levels refer to internal functions of the base monitor and are only relevant for service personnel. In addition, executing RMS with several active log levels will affect system performance. If log level 0 is specified, all possible log levels are activated. Valid log levels are listed in Table 3.

Log Level	Meaning
0	Turn on all log levels
1	Unused
2	Turn on detector tracing
3	Unused
4	Turn on `mskx` tracing (stack tracing of the base monitor)

Table 3: Log levels

Log Level	Meaning
5	Error or warning message
6	Heartbeats
7	Base monitor level
8	Detector error
9	Administrative command message
10	Basic-type level
11	Dynamic reconfiguration contracting level
12	Unused
13	Token level
14	Detector message
15	Local queue level
16	Local queue level
17	Script level
18	`userApplication` contract level
19	Temporary debug traces
20	`SysNode` traces
21	Message level
22	bm tracelog

Table 3: Log levels

Detector logging

If the log level is 2 (detector tracing) or higher, you can control detector logging with PCS or the RMS Wizard Tools. This is described in the section "RMS detector logging" on page 28.

2.7   Interpreting log files

Each process that makes up RMS generates three types of log messages: user, trace, and error. These log messages are contained in the following files:

switchlog   Records RMS events relevant to the user, such as switch requests and fault indications.

*<program>*log   Records trace messages or error messages for *program*. For example, messages from bm, the base monitor, are recorded in bmlog. The prefix for trace messages is as follows:
time:*file*:*line*:.
The prefix for error messages is as follows:
time:*file*:*line*:ERROR

switchlog file

The switchlog file contains the following five message types:

● Informational messages (notices)

● Warning messages

● Error messages

● Fatal error messages

● Output from scripts run by RMS

The first four categories of messages all follow this format:

timestamp: (*error code*, *error number*): *message type*: *message*: *delimiter*

There is a colon-space (:) between each field of the message where the timestamp is defined as follows:

yyyy-mm-dd hh:*mm*:*ss.xxx*

Message type is defined as one of the following:

● NOTICE

● WARNING

● ERROR

● FATAL ERROR

Messages are any text generated by the RMS product. This text can contain one or more new lines. The delimiter is defined as a colon followed by a series of four equal signs (:====).

The last category of messages (output from scripts) follows no specific format and is merely the redirected standard output and standard error from all scripts defined within the RMS configuration file. For example:

```
2001-05-07 11:01:54.568: WARNING: InitScript does not exist.: ====
```

2.8   System log

The base monitor of RMS writes messages to the switchlog file and also writes the same messages to the system log. By default, all the RMS messages go to both the switchlog file and also to the system log.

HV_SYSLOG_USE is an environment variable that you can modify so that messages will or will not show on the system log. If you do not want the messages to go into the system log, then set HV_SYSLOG_USE=0 in the hvenv.local file. Before changes can take effect, you must stop and restart RMS.

The default setting in hvenv is HV_SYSLOG_USE=1. This setting sends all RMS ERROR, FATAL ERROR, WARNING, and NOTICE messages to the system log and switchlog.

For Log3 RMS messages, the component number is 1080023.

hvlogcontrol

The hvlogcontrol utility prevents log files from becoming too large. Since large amounts of log files can take up disk space, hvlogcontrol limits the amount of log files to a specified amount set in one of the following environment variables selected by the system administrator:

● HV_LOG_ACTION_THRESHOLD

● HV_LOG_WARN_THRESHOLD

● HV_SYSLOG_USE

> **i** hvlogcontrol is called automatically from the crontab file, so there is no manual page.

2.9   PCS and Wizard Tools log files

The RMS configuration tools (PCS and the RMS Wizard Tools) log messages to files in the same log directory as is defined for RMS, according to the value set in the environment variable RELIANT_LOG_PATH. RMS configuration tools logging can be broken down into two categories as follows:

- Messages resource detectors
- All other messages

Detector logging will be explained in more detail in section "RMS detector logging" on page 28.

Unlike RMS, which logs most of its messages in the switchlog file, the RMS configuration tools log everything at an application level. All messages associated with a particular configured application are logged in the file *<RELIANT_LOG_PATH>*/*<application_name>*.log. The file is created when either offline or online processing for the application begins.

Each RMS configuration tools process generates the following two types of log messages:

- User
- Debug

The log messages are contained in the following files:

- switchlog—Records RMS events relevant to the user such as switch requests and fault indications. Also, the standard subapplications supplied by PCS and the Wizard Tools record detector reports concerning fault and offline transitions into the switchlog file.

- *<application_name>*.log—The application-specific log file records all messages associated with that application. The output from all scripts run by the application go into the log file.

- hvdet_*<xxx>*.g*<n>*log and pcsdet_*<xxx>*.g*<n>*log—These are detector log files which record all relevant information regarding the resources they are monitoring, such as all state transitions.

The format of most RMS configuration tools messages is as follows:

resource_ name:*state*:*timestamp*:*message_type*:*Message*:*delimiter*

There is a colon (:) between each field of the message.

The *resource_name* field is the name of the particular resource node in the RMS graph whose script is running. This field may be empty if no resource is associated with the message.

The *state* field is an indication of the type of action that is being performed, and is the value as set by RMS in the environment variable HV_SCRIPT_TYPE. The field typically contains the values online or offline. The RMS configuration tools also set the field with the value PreCheck, when a PreCheck script is being run. This field will be empty for messages of type DEBUG being printed.

The *timestamp* field contains the date when the message occurred and is written in the format *yyyy:mm:dd hh:mm:ss.xxx*, where *yyyy* is the 4 digit year; *mm* is the month number; *dd* is the day of the month; *hh* is the hour in the range of [0–23]; *mm* is the minute of the hour; *ss.xxx* is the number of seconds and milliseconds past the hour.

Message type is defined as one of the following:

● DEBUG

● NOTICE

● WARNING

● ERROR

● FATAL ERROR

Messages are any text generated by the RMS Wizard product. This text can contain one or more new lines. The delimiter is defined as a series of four equal signs (====).

Debug messages from scripts which are run can be forced by setting the environment variable HV_SCRIPTS_DEBUG to 1 in the hvenv.local file. The entry should appear as follows:

```
export HV_SCRIPTS_DEBUG=1
```

To turn off debug output, either remove the HV_SCRIPTS_DEBUG entry from the hvenv.local file, comment it out, or set the value to 0.

When debugging problems, the switchlog file as well as the application-specific log file, and any appropriate detector log files may all need to be viewed and interpreted.

2.9.1 RMS detector logging

The RMS detectors log information to both the `switchlog` file and to their own detector log files, which are located in *RELIANT_LOG_PATH* and have names of the form *<detector_name>*.`g`*<n>*`log`, where *<detector_name>* is the name of the detector and *<n>* is the instance number of the detector. For instance, the log file for the first (or only) *pcsdet_system* detector is *<RELIANT_LOG_PATH>*/`pcsdet_system.g1log`. Note that PCS detectors are named `pcsdet_`*xxx*, and Wizard Tools detectors are named `hvdet_`*xxx*.

All resource state changes are logged both to the `switchlog` file and to their own detector log file. Other detector messages are not logged to the `switchlog` file. A detector log file is created for each instance of a detector running.

Each detector maintains an internal 10 KB memory for logging debugging messages which are then printed out to the log file when an unexpected resource status report occurs. The buffer is a circular buffer such that if it fills before anything is printed out, it will be reused from the beginning and any existing data contained within the buffer will be overwritten and lost.

Each internal log message in the detector has an associated logging level. Only those messages which are less than or equal to the current log level setting will be added into the internal circular buffer. By default, only the internal messages marked with a debugging level of 1 are inserted into the buffer. The greater the value, the more debugging information is printed; however, the contents of logs may vary from detector to detector.

The circular buffer stores earlier messages, so when the buffer is written out, the log file will contain these earlier debug messages. They will have timestamps prior to the last reported item, which is the one that actually caused the buffer to be printed out.

Use the following procedures to adjust the log level in PCS or the Wizard Tools.

> **i** To enable detector logging, you must set the base monitor log level to 2 (detector tracing) or higher. See "Specifying the log level" on page 22.

2.9.1.1 Setting the log level in PCS

The valid range of log levels is 1 to 4, and the default value is 1. The log level can be modified from the PCS GUI as follows:

1. Make sure you are in expert mode by selecting *Option –> Expert Mode*.

2. Click anything in the left-hand configuration tree and then select *Edit –> DetectorSettings*. The *Detector Settings* window appears (Figure 11).
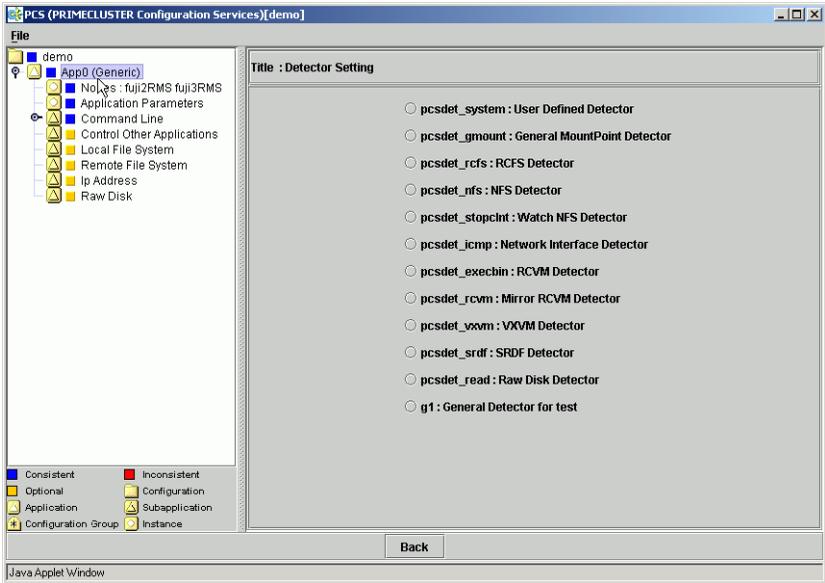
Figure 11: PCS Detector Details menu

3. Click the radio button of the detector you wish to modify, and you will advance to a window similar to the *pcsdet_system* setting shown in Figure 12.
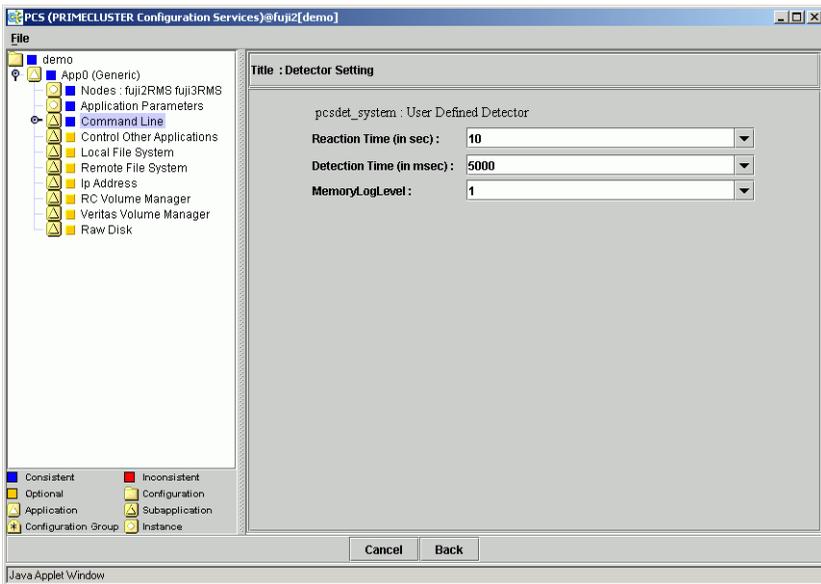
Figure 12: PCS Detector Details menu for pcsdet_system

4. Use the *MemoryLogLevel* drop-down list and change the value to the desired level. The log level applies only to the selected detector.

Alternatively, you can use the `pcsloglev` command with the following syntax:

`pcsloglev [-a]` *level*

where *level* is in the range 1 to 4. By default, the command sets the log level on the local node. Use the `-a` option to set the level on all nodes in the cluster.

2.9.1.2 Setting the log level in the Wizard Tools

The valid range of log levels is 1 to 9, and the default value is 1. The log level can be modified from the `hvw` command as follows:

1. Select the *Configuration-Edit-Global-Settings* menu.

2. Choose the *DetectorDetails* sub-menu. The screen in Figure 13 appears.

```
Detector Details:
 1) HELP                          12) ReactionTimeOfhvdet_nfs=10
 2) RETURN                        13) ReactionTimeOfhvdet_rcvm=33
 3) DEFAULTVALUES                 14) ReactionTimeOfhvdet_read=10
 4) ReactionTimeOfhvdet_ckhost=10 15) ReactionTimeOfhvdet_srdf=31
 5) ReactionTimeOfhvdet_ddm=19    16) ReactionTimeOfhvdet_stopclnt=10
 6) ReactionTimeOfhvdet_execbin=10 17) ReactionTimeOfhvdet_system=10
 7) ReactionTimeOfhvdet_glbassrt=10 18) ReactionTimeOfhvdet_vxvm=30
 8) ReactionTimeOfhvdet_gmount=10 19) ReactionTimeOfForeignDetectors=30
 9) ReactionTimeOfhvdet_icmp=10   20) MemoryLogLevel=1
10) ReactionTimeOfhvdet_locassrt=10 21) DynamicDetectorLogging=0
11) ReactionTimeOfhvdet_lvm=18
Set details for a detector:
```

Figure 13: Wizard Tools Detector Details menu

3. Select *MemoryLogLevel* and change the value to the desired level. The log level applies to all detectors.

2.9.2   Modifying log levels while RMS is running

It is possible to turn debug reporting on or off dynamically from the RMS Wizard detectors by using the hvw command as follows:

1. Select *Configuration-Edit-Global-Settings*.

2. Choose the *DetectorDetails* sub-menu.

3. Select the *DynamicDetectorLogging* menu item.

The default value is 0, which means that debugging is turned off. By setting the value to a number in the range 1 to 9, debugging is turned on. The greater the value, the more debug information that is generated; however, the volume of the generated log messages may vary from detector to detector. Note that modifying this setting will not take effect until the next time the configuration is activated.

Using hvw to turn on dynamic logging actually creates the file *<RELIANT_LOG_PATH>*/etc/wizardloglevel, which simply contains the desired debug level as a single-digit ASCII number. You can bypass the hvw command and create (or delete) the wizardloglevel file manually, as long as you follow these conventions:

● Log levels in the range 1 to 9 turn debugging on.

- If the file exists but is empty, a log level of 3 is assumed.

- If the file contains the number 0 (zero), or if the file does not exist, then debugging is turned off.

| i | Turning on the debugging should only be done when problems occur. Once the problems are resolved, debugging should be turned off to avoid filling the file system with extraneous information. |

2.10 PCS troubleshooting

2.10.1 PCS log and trace files

Experts and service personnel have two sources of information that may be useful in diagnosing PCS problems:

- The PCS log file is `/var/opt/SMAWpcs/log/pcs.log`. Basic diagnostic reports are written automatically to this file whenever PCS detects an internal inconsistency or error.

- Some problems may not be detected by the PCS self-diagnosis methods. In this case, use *Tools –> Trace* in the PCS GUI and check one of the trace level checkboxes; higher trace levels generate more information. Trace files are created in the `/var/opt/SMAWpcs/trace` directory.

| i | Trace information may generate large amounts of output. After a problem has been resolved, turn off tracing by unchecking the *Tools –> Trace* checkboxes, and remove the unneeded files from the `trace` directory. |

2.10.2 Manual Script Execution

The expert user will occasionally want to execute an object's scripts (online, offline etc.) one at a time for diagnostic purposes. This functionality is referred to as Manual Script Execution (MSE).

MSE is called from different places in the PCS GUI and PCS CUI:

- In the PCS GUI this functionary is available from the *Draw Graph* window.

- In the PCS CUI this functionality is available in the *Advanced Menu* under the Manual Script Execution Tree.

The configuration must be generated for MSE to available. To invoke MSE the user must right-click on the desired RMS resource in the PCS GUI graph, or select the MSE menu item in the PCS CUI, and then select the desired script to execute. The output of the scripts is displayed in the log file `/var/opt/SMAWpcs/log/pcs.log`.

2.11   RMS troubleshooting

When problems occur, RMS prints out meaningful error messages that will assist you in troubleshooting the cause. If no message is available, the following information may help you diagnose and correct some unusual problems:

● RMS dies immediately after being started.

At startup, the RMS base monitor exchanges its configuration checksum with the other base monitors on remote nodes. If the checksum of the starting base monitor matches the checksums from the remote nodes, the startup process continues. If the checksums do not match, then the RMS base monitor shuts down if all of the following conditions are true:

1. The base monitor has encountered a different checksum from a remote monitor within the initial startup period (defined by `HV_CHECKSUM_INTERVAL`).

2. There are no applications on this node that are online, waiting, busy, or locked.

3. There are no online remote base monitors encountered by this base monitor.

Otherwise, the base monitor keeps running, but all remote monitors whose checksums do not match the local configuration checksum are considered to be offline. Therefore, no message exchange is possible with these monitors, and no automatic or manual switchover will be possible between the local monitor and these remote monitors.

When different checksums are encountered, certain messages are placed in the switchlog explaining the situation.

$\boxed{\mathbf{i}}$ Configuration checksum differences often result when global environment variables are changed manually but inconsistently on different nodes.

Action:

To verify that a configuration checksum difference is not the cause of the problem, ensure that all nodes have been updated with the proper configuration by using the following procedure:

1. Stop RMS everywhere in the cluster. This may require an explicit `hvshut` command on any nodes that are not communicating with the others.

2. Determine which configuration to run. Use '`hvdisp -a`' or '`hvdisp -T SysNode`' on each node to verify the name of the configuration file. (The `hvdisp` command does not require root privilege.)

   A configuration may have the same name but different contents on two or more nodes if one of the following has occurred:

   – The most recent activation did not distribute the configuration to all nodes in the cluster. For instance, the user may have specified *Ignore Down Nodes* when activating with PCS.

   – If PCS or the Wizards Tools were used on more than one node in the cluster, there may be more than one version of the configuration with the same name.

3. Activate the correct configuration with the same tool (PCS or Wizard Tools) that was used to create it. For the correct procedure, see the *RMS Configuration and Administration Guide* for PCS or the Wizard Tools.

   Alternatively, redistribute the existing *<configname>*`.us` file with one of the following methods:

   – In the RMS Wizard Tools, use *Configuration Push*.

   – In PCS, use *Tools –> Activate –> Update*.

   Make sure the activation is successful so that all nodes are updated.

4. Start RMS on the entire cluster. All nodes should now be running the same configuration.

● RMS hangs after startup (processes are running, but `hvdisp` hangs)

This problem might occur if the local node is in the CF state LEFTCLUSTER from the point of view of one or more of the other nodes in the cluster.

Action:

Verify the problem by using 'cftool -n' on all cluster nodes to check for a possible LEFTCLUSTER state.

Use 'cftool -k' to clear the LEFTCLUSTER state. RMS will continue to run as soon as the node has joined the cluster. No restart should be necessary.

● RMS loops (or even dies) shortly after being started.

This problem could occur if the CIP configuration file /etc/cip.cf contains entries for the netmask. These entries are useless (not evaluated by CIP). From the RMS point of view these entries cannot be distinguished from IP addresses, which have the same format, so RMS will invoke a gethostbyaddr(). This normally does no harm, but in some unusual cases the OS may become confused.

Action:

Verify the problem by checking if netmask entries are present in /etc/cip.cf.

Remove the netmask entries, and restart RMS.

● RMS detects a node failure (network connection failed to host \...), but does not even attempt to kill the node.

This problem could occur if the failed node was already in a pending Wait state from an earlier failed kill request.

> **i** If a kill request fails, the SysNode remains in the Wait state until this state is manually cleared by the System Administrator.

Action:

Verify the problem by using 'hvdisp -T SysNode' to see the states of all SysNode objects. (The hvdisp command does not require root privilege.)

If you verify that a SysNode is in a pending wait state, use 'hvutil -o *SysNode*' or 'hvutil -u *SysNode*'.

> ⚠️ **Caution**
>
> 'hvutil -u' causes the surviving node to assume that the SysNode is actually dead, and it will invoke a failover immediately. If the node is still active, this may cause data corruption.

> ⚠️ **Caution**
>
> 'hvutil -o' causes the surviving node to assume that the SysNode was alive the entire time. Therefore, it will continue assuming to be in sync with the remote SysNode. If this assumption is not true, this could cause unpredictable behaviors and, in a worst case scenario, data corruption.

● The RMS base monitor detects a loss of cluster heartbeat, but there is no indication as to the reason for the loss.

RMS automatically invokes a tool that provides diagnostic information for this event. The diagnostic tool performs the following actions:

– Invokes truss(1) on Solaris or strace(1) on Linux to trace the detector process

– Turns on full RMS and detector logging with the -l0 (lowercase "L", zero) option

– Gathers system and users times for the process

The truss(1)/strace(1) invocation and logging levels will be terminated after the number of seconds specified in the ScriptTimeout attribute. All information is stored in the switchlog file.

Note that user-specified operations such as detector tracing will continue on each node, even if it appears to have left the cluster. In extreme cases, turning on high detail trace levels may affect the performance of a node and contribute to delays in its base monitor heartbeat.

Action:

See the switchlog for the diagnostic information.

2.11.1 Manual script execution

The expert user will occasionally want to execute an object's scripts (online, offline etc.) one at a time for diagnostic purposes. The RMS Wizard Tools provide a method to accomplish this for any object in the configuration.

Begin at the *Main configuration menu* (Figure 14).

```
penguin2: Main configuration menu, current configuration: test
No RMS active in the cluster
 1) HELP                             10) Configuration-Remove
 2) QUIT                             11) Configuration-Freeze
 3) Application-Create               12) Configuration-Thaw
 4) Application-Edit                 13) Configuration-Edit-Global-Settings
 5) Application-Remove               14) Configuration-Consistency-Report
 6) Application-Clone                15) Configuration-ScriptExecution
 7) Configuration-Generate          16) RMS-CreateMachine
 8) Configuration-Activate          17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action:
```

Figure 14: Wizard Tools Main configuration menu

▶ Select 15 for *Configuration-ScriptExecution*.

The *ManualExecution: Application selection menu* opens (Figure 15).

```
ManualExecution: Application selection menu:
1) HELP
2) QUIT
3) TOP
4) RETURN
5) app1
Application Name: 5
```

Figure 15: ManualExecution: Application selection menu

▶ Select the application name. In this example, we will select 5 for *app1*.

The *ManualExecution: Resource selection menu* opens (Figure 16).

```
ManualExecution: Resource selection menu:
1) HELP                              5) SHELL
2) QUIT                              6) Machine000_app1
3) TOP                               7) ManageProgram000_Cmd_APP1
4) RETURN                            8) app1
Choose the resource name: 8
```

Figure 16: ManualExecution: Resource selection menu

▶ Select the resource. In this example, we will select 8 for *app1*.

The *ManualExecution: Script selection menu* opens (Figure 17).

```
ManualExecution: Script selection menu:
1) HELP
2) QUIT
3) TOP
4) RETURN
5) SHELL
6) PreCheckScript:hvexec~-p~app1~test
7) PreOnlineScript:rm~-f~/usr/opt/reliant/tmp/app1.goingof-
fline~;~if~[~\"$HV_INTENDED_STATE\"~=
~\"Online\"~];~then~touch~/usr/opt/reliant/tmp/app1.online;~fi
8) PreOfflineScript:hvenable~app1~ALL~;~rm~-
f~/usr/opt/reliant/tmp/app1.online~;~touch~/usr/opt/reliant/tmp/app1.goingoffline
9) OfflineDoneScript:rm~-f~/usr/opt/reliant/tmp/app1.goingoffline
Choose the script: 6
```

Figure 17: ManualExecution: Script selection menu

▶ Select the script to execute. In this example, we will select 6 for the *PreCheck-Script*

Log output for the selected script will appear in
/var/opt/SMAWRrms/log/*<application_name>*.log.

3   Non-fatal error messages

This chapter contains a detailed list of all non-fatal RMS error messages that appear in the switchlog. Most messages are accompanied by a description of the probable cause(s) and a suggested action to correct the problem. In some cases, the description or action is self-evident and no further information is necessary.

Some messages in the listings that follow contain words printed in *italics*. These words are placeholders for values, names, or strings that will be inserted in the actual message when the error occurs.

RMS error code description

A prefix in each message contains an error code and message number identifying the RMS component that detected the problem. You may need to provide this prefix to support engineers who are diagnosing your problem. The following list summarizes the possible error codes and the associated component:

- `ADC:` Admin configuration
- `ADM:` Admin, command, and detector queues
- `BAS:` Startup and configuration errors
- `BM:` Base monitor
- `CML:` Command line
- `CRT:` Contracts and contract jobs
- `CTL:` Controllers
- `CUP:` userApplication contracts
- `DET:` Detectors
- `GEN:` Generic detector
- `INI:` init script
- `MAS:` Master process
- `MIS:` Miscellaneous
- `QUE:` Message queues
- `SCR:` Scripts
- `SWT:` Switch requests (hvswitch command)
- `SYS:` SysNode objects
- `UAP:` userApplication objects
- `US:` us files
- `WLT:` Wait list
- `WRP:` Wrappers

3.1   ADC: Admin configuration

● (ADC, 1) Since this host *<hostname>* has been online for no more than *time* seconds and due to the previous error, it will shut down now.

> *time* is the value of the environment variable HV_CHECKSUM_INTERVAL, if set, or 120 seconds otherwise. This message could appear when the checksums of the configurations of the local and the remote host are different, no more than *time* seconds have elapsed, and one of the following is true:
> – When the remote host is joining the cluster, and all the applications on the local host are either Offline or Faulted. RMS exits with exit code 60.
> – The configuration for the local host does not include the remote host, but the configuration for the remote host does include the local host. The local host *hostname* will shut down with exit code 60.

> *Action:*
> The local and the remote hosts are running different configurations. Make sure that both of them are running the same configuration.

● (ADC, 2) Since not all of the applications are offline or faulted on this host *<hostname>*, and due to the previous error, it will remain online, but neither automatic nor manual switchover will be possible on this host until *<detector>* detector will report offline or faulted.

> The checksums of the configurations of the local and the remote host are different, no more than the number of seconds determined by the value of the environment variable HV_CHECKSUM_INTERVAL have passed, and not all of the applications are offline or faulted. RMS will continue to remain online, but neither automatic nor manual switchover will be possible on this host until the detector *detector* reports offline or faulted.

> *Action:*
> Make sure that both the local and the remote host are running the same configuration.

- (ADC, 3) Remote host *<hostname>* reported the checksum (*remotechecksum*) which is different from the local checksum (*localchecksum*).

    If the checksum of the configuration file reported by the remote host *<hostname>* is different from the checksum of the configuration file on the local host, this message will appear.

    Action:
    The most likely cause for this would be that the local host and the remote host are running configuration files that differ. Make sure that the local host and the remote host are running the same configuration file.

- (ADC, 4) Host *<hostname>* is not in the local configuration.

    This message is a result of the following problem: If the checksum reported by the remote host is different from that of the local host and if the configuration for the local host does not include the remote host's name, but the configuration for the remote host *hostname* includes the local host.

    Action:
    Make sure that the local and the remote host are running the same configuration.

- (ADC, 5) Since this host *<hostname>* has been online for more than *time* seconds, and due to the previous error, it will remain online, but neither automatic nor manual switchover will be possible on this host until *<detector>* detector will report offline or faulted.

    If the checksums of the configurations of the local and the remote host are different and if more than *time* seconds have elapsed since this host has gone online (*time* is the value of the environment variable HV_CHECKSUM_INTERVAL if set, or 120 seconds otherwise), then RMS prints the above message.

    Action:
    Make sure that all the hosts in the cluster are running the same configuration file.

- (ADC, 15) `Global environment variable` *<envattribute>* `is not set in hvenv file.`

    RMS was unable to set the global environment variable *<envattribute>* because it has not been set in `hvenv`. *envattribute* can be any one of the following: `RELIANT_LOG_LIFE`, `RELIANT_SHUT_MIN_WAIT`, `HV_CHECKSUM_INTERVAL`, `HV_LOG_ACTION_THRESHOLD`, `HV_LOG_WARNING_THRESHOLD`, `HV_WAIT_CONFIG` or `HV_RCSTART`. This will eventually cause RMS to exit with exit code 1.

    Action:
    Set the value of the environment variable to an appropriate value.

- (ADC, 17) *<hostname>* `is not in the Wait state, hvutil -u request skipped!`

    When 'hvutil -u' has been invoked on a node, if the `SysNode` for that node is not in the Wait State, then this message will appear (internal option).

    Action:
    If the 'hvutil -u' was issued prematurely, then reissue the command once the node has reached the Wait state.

- (ADC, 18) `Local environmental variable` *<envattribute>* `is not set up in hvenv file.`

    If one of the local environment variables *<envattribute>* is not set in `hvenv`, this message is printed. *envattribute* can be any one of the following: `SCRIPTS_TIME_OUT`, `RELIANT_INITSCRIPT`, `RELIANT_STARTUP_PATH`, `HV_CONNECT_TIMEOUT`, `HV_MAXPROC` or `HV_SYSLOG_USE`. This will eventually cause RMS to exit with exit code 1.

    Action:
    Set the value of *envattribute* to an appropriate value.

- (ADC, 20) *<hostname>* `is not in the Wait state. hvutil -o request skipped!`

    The 'hvutil -o' command has been invoked on a node, but its `SysNode` is not in the Wait State. (Internal option).

    Action:
    The 'hvutil -o' was issued prematurely. Reissue the command after the `SysNode` has reached the Wait state.

- (ADC, 25) Application *<userapplication>* is locked or busy, modification request skipped.

   hvmod has been invoked without the −l option, and the application is busy. Some other modification is already in progress, or some requests are being processed, or application contracting is ongoing.

   Action:
   Reissue the hvmod command when the application has completed the current switch request.

- (ADC, 27) Dynamic modification failed.

   Dynamic modification has failed. The exact reason for the failure is displayed in the message preceding this one.

   Action:
   Check the error messages occurring in the switchlog or prior to this message to find out the exact cause of the failure.

- (ADC, 30) HV_WAIT_CONFIG value *<seconds>* is incorrect, using 120 instead.

   If the value of the environment variable HV_WAIT_CONFIG is 0 or has not been set, the default value of 120 is used instead.

   Action:
   Set the value of HV_WAIT_CONFIG in /opt/SMAW/SMAWRrms/bin/hvenv.

- (ADC, 31) Cannot get the NET_SEND_Q queue.

   RMS uses the NET_SEND_Q queue for transmitting contract information. If there is some problem with this queue, the operation is aborted. The operation can be any one of the following: hvrcp, hvcopy.

   Action:
   Contact field support.

- (ADC, 32) Message send failed during the file copy of file *<file>*.

   A error occurred while transferring file *<file>* across the network.

   Action:
   Check if there are any problems with the network.

● (ADC, 33) Dynamic modification timeout.

  The time taken for dynamic modification is greater than the timeout limit.
  The timeout limit is the greater of the environment variable `MODIFYTIM-`
  `EOUTLIMIT` (if defined) or 0. If the variable is not defined, the default
  timeout limit is 120 seconds.

  Action:
  Contact field support.

● (ADC, 34) Dynamic modification timeout during start up — bm
  will exit.

  The time taken for dynamic modification during `bm` startup is greater than
  the timeout limit. The timeout limit is the greater of the environment
  variable `MODIFYTIMEOUTLIMIT` (if defined) or 0. If the variable is not
  defined, the default timeout limit is 120 seconds. RMS exits with exit code
  63.

  Action:
  Contact field support.

● (ADC, 35) Dynamic modification timeout, bm will exit.

  Critical internal error.

  Action:
  Contact field support.

● (ADC, 37) 75. Dynamic modification failed: cannot make a
  non-critical resource *<resource>* critical by changing its
  attribute MonitorOnly to 0 since this resource is not online
  while it belongs to an online application *<userapplication>*;
  switch the application offline before making this resource
  critical.

  During dynamic modification, if there is an attempt to make a non-critical
  resource *<resource>* `MonitorOnly` while it is not online and the appli-
  cation *<userapplication>* is Online this message is printed, and dynamic
  modification is aborted.

  Action:
  Switch the userApplication Offline before making the resource critical.

● (ADC, 38) 76. Dynamic modification failed: application *<userapplication>* has no children, or its children are not valid resources.

> If RMS finds that the userApplication *<userapplication>* will have no children while performing dynamic modification, this message is printed to the switchlog and dynamic modification is aborted.

> *Action:*
> Make sure that the userApplication has valid children while performing dynamic modification.

● (ADC, 39) The putenv() has failed (*failurereason*)

> The wizards use the environment variable HVMOD_HOST during dynamic modification. This variable holds the name of the host on which hvmod has been invoked. If this variable cannot be set with the function putenv(), then this message is printed to the switchlog along with the reason *failurereason*.

> *Action:*
> Check the reason *failurereason* in the switchlog to find out why this operation has failed and take corrective action based on this.

● (ADC, 41) The Wizard action failed (*command*)

> Wizards make use of an action file during hvmod. If the execution of this action file (*command*) has failed due to the process exiting by using an exit call, this message is printed to the switchlog along with the reason for this failure.

> *Action:*
> Check the switchlog for finding the reason for this failure and rectify it before reissuing the hvmod command.

● (ADC, 43) The file transfer for *<filename>* failed in "*command*". The dynamic modification will be aborted.

> During dynamic modification, files containing modification information are transferred between the hosts of the cluster. If, for any reason, a file transfer fails, the dynamic modification is aborted.

> *Action:*
> Make sure that host and cluster conditions are such that *command* can be safely executed.

● (ADC, 44) The file transfer for *\<filename\>* failed in
  "*command*". The join will be aborted.

  When a host joins a cluster, it receives a cluster configuration file. If, for
  any reason, a file transfer fails, the dynamic modification is aborted.

  Action:
  Make sure that host and cluster conditions are such that *command* can be
  safely executed.

● (ADC, 45) The file transfer for *\<filename\>* failed in "*command*"
  with errno *\<errno\>* − *errorreason*. The dynamic modification will
  be aborted.

  During dynamic modification, files containing modification information
  are transferred between the hosts of the cluster. If, for any reason, a file
  transfer fails, the dynamic modification is aborted. A specific reason for
  this failure is referred to by the OS error code ERRNO and its explanation
  in ERRORREASON.

  Action:
  Make sure that host and cluster conditions are such that *command* can be
  safely executed.

● (ADC, 46) The file transfer for *\<filename\>* failed with unequal
  write byte count, expected *expectedvalue* actual *actualvalue*. The
  dynamic modification will be aborted.

  During dynamic modification, files containing modification information
  are transferred between the hosts of the cluster. During the transfer, RMS
  keeps track of the integrity of the transferred data by counting the bytes
  transferred. This count can be incorrect if the transfer process is broken
  or interrupted.

  Action:
  Make sure that host, cluster and network conditions are such that
  command can be safely executed.

● (ADC, 47) RCP fail: can't open file *filename*.

  If the file *\<filename\>* that has been specified as the file to be copied from
  the local host to the remote host cannot be opened for reading, this
  message is printed.

Action:
Make sure that the file *<filename>* is readable.

● (ADC, 48) RCP fail: fseek errno *errno*.

During a file transfer between the hosts, RMS encountered a problem indicated by the OS error code ERRNO.

Action:
Make sure that the host, cluster and network conditions are such that file transfer proceeds without errors.

● (ADC, 49) Error checking hvdisp temporary file *<filename>*, errno *<errno>*, hvdisp process pid *<processid>* is restarted.

The RMS base monitor periodically checks the integrity and size of the temporary file used to transfer configuration data to the hvdisp process. If this file cannot be checked, then hvdisp process is restarted automatically, though some data may be lost and not displayed at this time. Specific OS error code for the error encountered is displayed in ERRNO.

Action:
Make sure that the host conditions are such that the temporary file can be checked. Sometimes, you may need to restart the hvdisp process by hand.

● (ADC, 57) An error occurred while writing out the RMS configuration for the joining host. The hvjoin operation is aborted.

When a remote host joins a cluster, this host attempts to dump its own configuration for a subsequent transfer to the remote host. If the configuration cannot be saved, the hvjoin operation is aborted.

Action:
One of the previous messages contain a detailed explanation about the error occurring while saving the configuration. Correct the host environment according to the explanation, or contact field support.

- (ADC, 58) `Failed to prepare configuration files for transfer to a joining host. Command used` *<command>*`.`

   When a remote host joins a cluster, this host attempts to prepare its own configuration for a subsequent transfer to the remote host. For that, it uses the command *<command>*. If the *<command>* fails, the `hvjoin` operation is aborted.

   Action:
   Contact field support.

- (ADC, 59) `Failed to store remote configuration files on this host. Command used` *<command>*`.`

   When this host joins a cluster, this host attempts to store remote configuration files for a subsequent dynamic modification on this host. For that, it uses the command *<command>*. If the *<command>* fails, the `hvjoin` operation is aborted.

   Action:
   Contact field support.

- (ADC, 60) `Failed to compress file` *<file>*`. Command used` *<command>*`.`

   File transfer is a part of some RMS operations such as dynamic modification and `hvjoin`. Before transferring a file *<file>* to a remote host, it must be compressed with the command *<command>*. If the *<command>* fails, the operation that requires the file transfer is aborted.

   Action:
   Contact field support.

- (ADC, 61) `Failed to shut down RMS on host` *<host>*`.`

   While performing RMS cluster-wide shutdown, RMS on host *<host>* failed to shut down.

   Action:
   Contact field support.

● (ADC, 62) Failed to shut down RMS on this host, attempting to exit RMS.

   While performing RMS clusterwide shutdown, RMS on this host failed to shut down. Another attempt to shut down this host is automatically initiated.

   Action:
   Contact field support.

● (ADC, 63) Error *<errno>* while reading file *<file>*, reason: *<reason>*.

   While reading file *<file>*, an error *<errno>* occurred explained by *<reason>*. File reading errors may occur during dynamic modification, or during hvjoin operation.

   Action:
   Contact field support.

● (ADC, 68) Error *<errno>* while opening file *<file>*, reason: *<reason>*.

   While opening file *<file>*, an error *<errno>* occurred explained by *<reason>*. File open errors may occur during dynamic modification.

   Action:
   Verify the file existence and reissue dynamic modification request.

● (ADC, 70) Message sequence # is out of sync — File transfer of file *<filename>* has failed.

   Critical internal error.

   Action:
   Contact field support.

3.2 ADM: Admin, command, and detector queues

- (ADM, 3) 31. Dynamic modification failed: some resource(s) supposed to come offline failed.

  During dynamic modification when new resource(s) that are to be added to a parent object that is offline cannot be brought offline, this message is printed.

  Action:
  Make sure the new resource(s) can be brought to the offline state and reissue the `hvmod` command.

- (ADM, 4) 30. Dynamic modification failed: some resource(s) supposed to come online failed.

  During dynamic modification when new resource(s) that are to be added to a parent object that is online by executing the online scripts cannot be brought online, dynamic modification is aborted.

  Action:
  Make sure the new resource(s) can be brought to the online state and reissue the `hvmod` command.

- (ADM, 5) 17. Dynamic modification failed: object *<object>* is not linked to any application.

  During dynamic modification, if there is an attempt to add an object *<object>* that does not have a parent (and hence not linked to any `userApplication`), this message is printed and dynamic modification is aborted.

  Action:
  Make sure that every object being added during dynamic modification is linked to a `userApplication`.

- (ADM, 6) 36. Dynamic modification failed: cannot add new resource *<resource>* since another existing resource with this name will remain in the configuration.

  When RMS receives a directive to add a new resource *<resource>* with the same name as that of an existing resource, this message is printed to the switchlog and dynamic modification aborts.

Action:
> Make sure that when adding a new resource, its name does not match the name of any other existing resource.

● (ADM, 7) 35. Dynamic modification failed: cannot add new resource <*resource*> since another existing resource with this name will not be deleted.

> When RMS receives a directive to add a new resource <*resource*> with the name of an existing resource, it prints out this message and dynamic modification aborts.

Action:
> Make sure that when adding a new resource, its name does not match the name of any other existing resource.

● (ADM, 8) 29. Dynamic modification failed: cycle of length <*cyclelength*> detected in resource <*resource*> -- <*cycle*>.

> In the overall structure of the graph of the RMS resources, no cycles are allowed along the chains of parent/child links. If this is not the case then dynamic modification fails and the message specified above will be printed to the switchlog.

Action:
> Get rid of the cycles.

● (ADM, 9) 34. Dynamic modification failed: cannot modify resource <*resource*> since it is going to be deleted.

> Since, deleting a resource causes all its children with no other parents to get deleted as well, deleting a resource and then modifying the attributes of the deleted resource or a child of that resource that has no other parents leads to dynamic modification being aborted and the message being printed to the switchlog.

Action:
> While performing dynamic modification of a resource make sure that the resource that is being modified has not been deleted.

- (ADM, 11) 37. Dynamic modification failed: cannot delete object *<resource>* since it is a descendant of another object that is going to be deleted.

  When there is an attempt to delete a child object when the parent object has been deleted, the above message will appear in the switchlog and dynamic modification aborted.

  Action:
  Make sure that when an object is being deleted explicitly, its parents have not already been deleted because that means this object has also been deleted.

- (ADM, 12) 38. Dynamic modification failed: cannot delete *<resource>* since its children will be deleted.

  When there is an attempt to delete a resource *<resource>* whose children have already been deleted, the above message will appear in the switchlog and dynamic modification aborted.

  Action:
  Make sure that when a resource is being deleted explicitly, its children have not already been deleted.

- (ADM, 13) 52. dynamic modification failed: object *<resource>* is in state *<state>* while needs to be in one of stateOnline, stateStandby, stateOffline, stateFaulted, or stateUnknown.

  Every resource has to be in either one of the states: stateOnline, stateOffline, stateFaulted, stateUnknown or stateStandby. If the resource *<resource>* is not in any of the states mentioned above, it prints the above message and dynamic modification is aborted. Theoretically this is not possible.

  Action:
  Contact field support.

- (ADM, 14) 48. Dynamic modification failed: cannot link to or unlink from an application *<userapplication>*.

  If the parent of the resource is a userApplication, then linking to or unlinking a child from that parent is not possible. If there is an attempt to perform this, then the above message will be printed to the switchlog and dynamic modification will be aborted.

Action:

    Do not link or unlink a resource from a `userApplication`.

● `(ADM, 15) 41. Dynamic modification failed: parent object` *<parentobject>* `is not a resource.`

    When RMS gets a directive to link existing resources during dynamic modification, and the parent object *<parentobject>* to which the child object is being linked is not a resource, then dynamic modification fails, and this message is printed.

Action:

    Make sure that while linking 2 objects, the parent of the child object is a resource.

● `(ADM, 16) 42. Dynamic modification failed: child object` *<childobject>* `is not a resource.`

    When RMS gets a directive to link existing resources during dynamic modification, if the child object *<childobject>* that is being linked to a parent object is not a resource, then dynamic modification fails and this message is printed.

Action:

    Make sure that while linking 2 objects, the child of the parent object is a resource.

● `(ADM, 17) 43. Dynamic modification failed: cannot link parent` *<parentobject>* `and child` *<childobject>* `since they are already linked.`

    An attempt was made to link a parent *<parentobject>* and a child *<childobject>* that are already linked. This message is printed, and dynamic modification is aborted.

Action:

    While trying to perform dynamic modification, make sure that the parent and the child that are to be linked are not already linked.

● (ADM, 18) 49. `Dynamic modification failed: cannot link a` `faulted child` <*childobject*> `to parent` <*parentobject*> `which is not` `faulted.`

> While creating a new link between 2 existing objects, during dynamic modification, a faulted child <*childobject*> cannot be linked to a parent <*parentobject*> that is not faulted. The child first needs to be brought to the state of the parent. If this condition is violated, the aforementioned message will be printed to the switchlog. Dynamic modification is aborted.

Action:
> Bring the faulted child to the state of the parent before linking them.

● (ADM, 19) 50. `Dynamic modification failed: cannot link child` <*childobject*> `which is not online to online parent` <*parentobject*>`.`

> While linking 2 existing objects during dynamic modification, the combination of states parent Online and child not Online is not allowed. When this happens, dynamic modification is aborted and a message is printed to the switchlog.

Action:
> The child <*childobject*> first needs to be brought to the online state before linking it to the online parent <*parentobject*>.

● (ADM, 20) 51. `Dynamic modification failed: cannot link child` <*childobject*> `which is neither offline nor standby to offline` `or standby parent` <*parentobject*>`.`

> Any attempt to link 2 existing objects in which the child is neither in the Offline nor the Standby state, and the parent is in the Offline or Standby state, is prohibited. This message is printed in the switchlog, and dynamic modification is aborted.

Action:
> The child needs to be first brought to offline or standby state before linking it to the parent that is in offline or standby state.

● (ADM, 21) 44. `Dynamic modification failed: Cannot unlink` `parent` *<parentobject>* `and child` *<childobject>* `since they are not` `linked.`

   Trying to unlink object *<parentobject>* from object *<childobject>* when they are not already linked results in this message with dynamic modification aborted.

   Action:
   If you want to unlink 2 objects make sure that they share a parent child relationship.

● (ADM, 22) 46. `Dynamic modification failed: child` *<childobject>* `will be unlinked but not linked back to any of the applica-` `tions.`

   Unlinking a child *<childobject>* so that no links remain linking it to any `userApplication` is not allowed.

   Action:
   Make sure that the child is still linked to a `userApplication`.

● (ADM, 23) 47. `Dynamic modification failed: sanity check did` `not pass for linked or unlinked objects.`

   Dynamic modification performs some sanity checks to ensure that all of the following are true:
   – The `HostName` attribute is present only for children of `userApplication` objects.
   – The child of a `userApplication` does not have another parent.
   – Each object belongs to only one `userApplication`.
   – Leaf objects have detectors.
   – Leaf objects that have the `DeviceName` attribute have it set to a valid value.
   – The length of the attribute `rName` for the leaf objects is smaller than the maximum.
   – There are no duplicate lines in the `hvgdstartup` file.
   – The `kind` argument for the detector in the `hvgdstartup` is specified.
   – All detectors can be loaded.
   – A valid value has been specified for the `rKind` attribute.
   – The `ScriptTimeout` value is greater than the detector cycle time.
   – No objects are `and` and `or` at the same time.
   – `ClusterExclusive` and `LieOffline`, which are mutually exclusive, are not used together.

If some of these sanity checks fail, then this message will be printed and dynamic modification is aborted. A FATAL message is also printed to the switchlog with more details as to why the sanity check failed.

Action:
Make sure that the sanity checks mentioned above pass.

● (ADM, 24) 45. Dynamic modification failed: object *<object>* that is going to be linked or unlinked will be either deleted, or unlinked from all applications.

Any attempt to perform the operations of deleting an object *<object>* from the RMS resource graph and then trying to unlink it from its parent object or vice versa results in dynamic modification being aborted and the above message being printed to the switchlog.

Action:
Make sure that the operations of deletion and unlinking are not performed on an object at the same time.

● (ADM, 25) 1. Dynamic modification failed: parent object *<parentobject>* is absent.

When a new object is being added to an existing configuration, it should have an existing object *<parentobject>* as its parent, if not then, dynamic modification is aborted and the message is printed to the switchlog.

Action:
Make sure that the parent specified for a new object that is being added is existent.

● (ADM, 26) 18. Dynamic modification failed: parent object *<parentobject>* is neither a resource nor an application.

When a new object is being added to an existing configuration, if the parent object *<parentobject>* that has been specified is not a resource, it leads to dynamic modification aborting and the message being printed. Dynamic modification is aborted.

Action:
Make sure that the parent object specified for a new object is a resource.

● (ADM, 27) 2. Dynamic modification failed -- child object
  <childobject> is absent.

  Any attempt to link to a child object *<childobject>* that is non-existent
  leads to this message and dynamic modification aborts.

  Action:
  Make sure that the child object to be linked to exists.

● (ADM, 28) 19. Dynamic modification failed: child object
  <childobject> is not a resource.

  When a new object *<childobject>* being added to an existing configuration
  is not a resource, this message is printed, and dynamic modification is
  aborted.

  Action:
  Make sure that the child object specified is a resource.

● (ADM, 29) 3. Dynamic modification failed -- parent object
  <parentobject> is absent.

  Action:
  Critical error. Contact field support.

● (ADM, 30) 20. Dynamic modification failed: parent object
  <parentobject> is not a resource.

  During dynamic modification if there is a request to add a new parent
  object *<parentobject>* that is not a resource, this message is printed, and
  dynamic modification aborts.

  Action:
  Make sure that the object being added as a parent object is a resource.

● (ADM, 31) 4. Dynamic modification failed: child object
  <childobject> is absent.

  As part of dynamic modification, if the specified child object *<childobject>*
  does not exist, then this message is printed, and dynamic modification is
  aborted.

  Action:
  Make sure that the child object that has been specified exists.

- (ADM, 32) 21. `Dynamic modification failed: child object` *<childobject>* `is not a resource.`

  When adding a new object to the RMS resource graph, if the child *<childobject>* of this new object is not a resource, dynamic modification aborts.

  Action:
  Make sure that when adding a new object, its child is a resource.

- (ADM, 33) 5. `Dynamic modification failed: object` *<object>* `cannot be deleted since either it is absent or it is not a resource.`

  If RMS gets a directive to delete an object *<object>* that is either non-existent or not a resource, this message is printed along with the failure of dynamic modification.

  Action:
  Make sure that you don't try to delete an object that does not exist.

- (ADM, 34) 22. `Dynamic modification failed: deleted object` *<object>* `is neither a resource nor an application nor a host.`

  An object deleted during dynamic modification is neither a resource type object, nor a `userApplication` nor a `SysNode` object. Only resources, applications and hosts (`SysNode` objects) can be deleted during dynamic modification.

  Action:
  Do not delete this object, or delete another object.

- (ADM, 37) 6. `Dynamic modification failed: resource` *<object>* `cannot be brought online and offline/standby at the same time.`

  When a resource *<object>* is added to an existing RMS resource graph and it is linked as a child to two parent objects, one of which is online and the other offline/standby, this message is printed: a child object needs to be brought to the state of its parent.

  Action:
  Make sure that both the parents of the resource to be added are in the same state before adding it.

- (ADM, 38) 7. Dynamic modification failed: existing parent resource *<parentobject>* is in state *<state>* but needs to be in one of stateOnline, stateStandby, stateOffline, state-Faulted, or stateUnknown.

  During dynamic modification, if the state *<state>* of a parent resource *<parentobject>* is not one of the states stateOnline, stateOffline, state-Faulted, or stateUnknown, dynamic modification aborts.

  Action:
  Make sure that the state of the parent resource is one of the states mentioned above.

- (ADM, 39) 28. Dynamic modification failed: new resource *object* which is a child of application *<userapplication>* has its HostName *<hostname>* the same as another child of application *<userapplication>*.

  When a new object *object* is being added as a child of *<userapplication>* and the value of its HostName attribute is the same as the value of the HostName attribute of an existing child of *<userapplication>*, this message is printed, and dynamic modification is aborted.

  Action:
  Make sure that the HostName attribute of an object that is being added to userApplication is different from the values of the HostName attributes of other first level children of *userapplication*.

- (ADM, 40) 25. Dynamic modification failed: a new child *<childobject>* of existing application *<userapplication>* does not have its HostName set to a name of any sysnode.

  When a new child object *<childobject>* is added to an application *<userapplication>* during dynamic modification, if the HostName attribute is missing for this object, this message is printed, and dynamic modification is aborted.

  Action:
  The first level object under *userapplication* must have a HostName attribute.

● (ADM, 41) 8. Dynamic modification failed: existing child *<childobject>* is not online, but needs to be linked with *<parentobject>* which is supposed to be brought online.

    If both the parent *<parentobject>* and the child *<childobject>* have detectors associated with them, if the state of the child is not online, but it needs to be linked to the parent that is supposed to be online, then this message will be printed and dynamic modification aborted.

    Action:
    Make sure that the parent and the child are in a similar state.

● (ADM, 42) 9. Dynamic modification failed: existing child *<childobject>* is online, but needs to be linked with *<parentobject>* which is supposed to be brought offline.

    Trying to link a child *<childobject>* that is online to a parent object, which is supposed to go offline, is not allowed, and dynamic modification is aborted.

    Action:
    Make sure that the parent and the child are in a similar state.

● (ADM, 43) 10. Dynamic modification failed: linking the same resource *<childobject>* to different applications *<userapplication1>* and *<userapplication2>*.

    When RMS gets a directive to add a new child object *<childobject>* having as parent and child resources belonging to different applications *<userapplication1>* and *<userapplication2>*, the above message is printed and dynamic modification aborts.

    Action:
    When adding a new resource make sure that it does not have as its parent and children, resources belonging to different applications.

● (ADM, 44) 11. Dynamic modification failed: object *<object>* does not have an existing parent.

    Any attempt to create an object *<object>* that does not have an existing parent leads to this message and dynamic modification aborts.

    Action:
    Make sure that the object *<object>* has an existing object as its parent.

- (ADM, 45) 55. Dynamic modification failed: HostName is absent or invalid for resource *<object>*.

    If the HostName attribute of object *<object>* is an invalid value then this message occurs and dynamic modification is aborted. If the HostName attribute is missing, (ADM, 40) will take care of it.

    Action:
    Set the HostName attribute of resource *<object>* to the name of a valid SysNode.

- (ADM, 46) 12. Dynamic modification failed: linking the same resource *<object>* to different applications *<userapplication1>* and *<userapplication2>*.

    RMS received a directive to add a new child object *<object>* by linking it to parent objects belonging to different applications *<userapplication1>* and *<userapplication2>*. Dynamic modification is aborted.

    Action:
    When adding a new child resource, make sure that it does not have as its parents resources belonging to different applications.

- (ADM, 47) 23. Dynamic modification failed: parent object *<parentobject>* belongs to a deleted application.

    Any attempt to add a new node having as its parent *<parentobject>* fails if the parent *<parentobject>* is the child of an object that has been deleted, because deleting an object automatically causes its children to be deleted as well if they don't have any other parents. This causes dynamic modification to fail.

    Action:
    When adding a new object make sure that its parent has not already been deleted.

- (ADM, 48) 24. Dynamic modification failed: child object *<childobject>* belongs to a deleted application.

    Any attempt to delete an object *<childobject>* belonging to a deleted application elicits this response from RMS because deleting an application automatically causes all its children to be deleted as well.

    Action:
    Do not try to delete an object belonging to an already deleted application.

● (ADM, 49) 24. `Dynamic modification failed: deleted object` *<objectname>* `belongs to a deleted application.`

  Any attempt to delete an object *<objectname>* that belongs to a deleted application leads to this error because deleting an application deletes all its children including *<objectname>*.

  Action:

  Make sure that before an object is deleted, it does not belong to an application that is being deleted.

● (ADM, 50) 40. `Dynamic modification failed: cannot delete object` *<object>* `since it is a descendant of a new object.`

  When RMS gets a directive to delete an object *<object>*, which is a descendant of a new object, this message is printed, and dynamic modification is aborted.

  Action:

  Make sure that when an object is being deleted, it is not a descendant of a new object.

● (ADM, 51) 15. `Dynamic modification failed: cannot link to child` *<childobject>* `since it will be deleted.`

  When RMS gets a directive to link to a child *<childobject>* that is going to be deleted, dynamic modification aborts.

  Action:

  Do not link to a child object that is to be deleted.

● (ADM, 52) 16. `Dynamic modification failed: cannot link to parent` *<parentobject>* `since it will be deleted as a result of deletion of object` *<object>*.

  If there is an attempt to delete an object *<object>* and use its descendants (which should be deleted as a result of deleting the parent) as the parent for a new resource that is being added to the RMS resource graph, this error message is printed and dynamic modification aborts.

  Action:

  Do not attempt to delete an object and use its descendant as the parent for a new resource.

- (ADM, 53) 26. Dynamic modification failed: *<node>* is absent.

  An attempt was made to modify the attribute of a node *<node>* that is absent. This message is printed and dynamic modification is aborted.

  Action:
  Modify the attributes of an existing node.

- (ADM, 54) 27. Dynamic modification failed: NODE *<object>*, attribute *<attribute>* is invalid.

  When RMS receives a directive to modify a node *<object>* with attribute *<attribute>* that has an invalid value, this message is printed, and dynamic modification is aborted.

  Action:
  Specify a valid value for the attribute *<attribute>*.

- (ADM, 55) Cannot create admin queue.

  RMS uses Unix queues internally for interprocess communication. Admin queue is one such queue that is used for communication between RMS and other utilities like `hvutil`, `hvmod`, `hvshut`, `hvswitch` and `hvdisp`. If RMS cannot create this queue due to some reason, RMS exits with exit code 50.

  Action:
  Restart RMS

- (ADM, 57) hvdisp — open failed — *filename*.

  If RMS is unable to open the file `/opt/SMAW/SMAWRrms/locks/.rms.`*<pid>* for writing when `hvdisp` has been invoked, this message is printed.

  Action:
  Verify that the directory `/opt/SMAW/SMAWRrms/locks` exists and allows files to be created (correct permissions, free space in the file system, free inodes). If one of these problems exists, fix it via the appropriate administrator operation. If none of these problems apply, but the RMS failure still occurs, contact RMS support.

● (ADM, 58) hvdisp – open failed – *filename* : *errormsg*.

When hvdisp is unable to open the file *file*
(/opt/SMAW/SMAWRrms/locks/.rms.*<pid>*) for writing, it prints out the
reason *errormsg*.

Action:

Verify that the directory /opt/SMAW/SMAWRrms/locks exists and allows
files to be created (correct permissions, free space in the file system, free
inodes). If one of these problems exists, fix it via the appropriate admin-
istrator operation. If none of these problems apply, but the RMS failure
still occurs, contact RMS support.

● (ADM, 59) *userapplication*: modification is in progress, switch
request skipped.

This message is printed to the switchlog because commands like
hvswitch, hvutil and hvshut cannot run in parallel with a non local
hvmod.

Action:

Make sure that before a hvswitch is performed, hvmod is not operating
on *userapplication*.

● (ADM, 60) *<resource>* is not a userApplication object, switch
request skipped!

While performing a switch, hvswitch requires a userApplication as
its argument. If the resource *<resource>* is not a userApplication, this
message is printed.

Action:

Check the man page for hvswitch for usage information.

● (ADM, 62) The attribute <ShutdownScript> may not be
specified for object *<object>*.

The attribute ShutdownScript is a hidden attribute of a SysNode. The
RMS base monitor automatically defines its value -- users cannot change
it in any way.

Action:

Do not attempt to change the built-in value of the ShutdownScript
attribute.

● (ADM, 63) System name *<sysnode>* is unknown.

This message can occur in these scenarios:
– The name of the SysNode specified in hvswitch is not included in the current configuration. ('hvswitch [-f] *userapplication* [*sysnode*]')
– The name of the SysNode specified for 'hvshut -s *sysnode*' is not a valid one, i.e., *sysnode* is not included in the current configuration.
– The name of the SysNode specified for 'hvutil -ou' is unknown (hidden options).

Action:
Specify a SysNode that is included in the current configuration, i.e., appears in the *configname*.us file.

● (ADM, 67) *sysnode* Cannot shut down.

This message could appear if 'hvshut -a' was invoked and not all of the nodes replied with an acknowledgement.

Action:
Login to the remote hosts. If RMS is still running, perform 'hvutil -f *<userapplication>*' to shut down each application one at a time. If this fails, refer to the switchlog and *<userapplication>*log files to find the reason for the problem. If all applications have been shut down correctly, perform a forced RMS shutdown with 'hvshut -f'. Report the problem to RMS support.

● (ADM, 70) NOT ready to shut down.

The node on which 'hvshut -a' has been invoked is not yet ready to be shut down because the application is busy on the node.

Action:
Wait until the ongoing action (e.g., switchover, dynamic reconfiguration) has terminated.

● (ADM, 75) 57. Dynamic modification failed: child *<resource>* of userApplication object *<userapplication>* has HostName attribute *<hostname>* common with other children of the same userApplication.

This message occurs if the RMS internal sanity-check functions detect a severe configuration problem. This message should not occur if the configuration has been set up using RMS configuration wizards.

Action:
> Contact field support.

● (ADM, 76) Modification of attribute <*attribute*> is not allowed within existing object <*object*>.

> The attribute <*attribute*> is constant and can only be set in a configuration file.

Action:
> Make sure that there is no attempt to modify <*attribute*> within <*object*>.

● (ADM, 77) 58. Dynamic modification failed: cannot delete object *object* since its state is currently being asserted.

> This message can appear in the switchlog if dynamic modification is being performed on an object that is being asserted.

Action:
> Perform the modification after the assertion has been fulfilled.

● (ADM, 78) 59. Dynamic modification failed: PriorityList <*prioritylist*> does not include all the hosts where the application <*userapplication*> may become Online. Make sure that PriorityList contains all hosts from the HostName attribute of the application's children.

> Set PriorityList for <*userapplication*> to include all the host names from the HostName attribute of the application's children.

Action:
> No duplicate host names should be present in the PriorityList.

● (ADM, 79) 60. Dynamic modification failed: PriorityList <*prioritylist*> includes hosts where the application <*userapplication*> may never become Online. Make sure PriorityList contains only hosts from the HostName attributes of the application's children.

> The HostName attribute of one or more of the children specifies hosts that are not in the parent's PriorityList attribute.

Action:

> Set the `PriorityList` attribute of *<userapplication>* to include all the host names listed in the `HostName` attributes of the application's children. No duplicate host names should be present in the `PriorityList`.

● (ADM, 81) 61. Dynamic modification failed: application *<userapplication>* may not have more than *<maxcontroller>* parent controllers as specified in its attribute MaxControllers.

> If *<userapplication>* uses more parent controllers than specified by the attribute `MaxControllers` (*<maxcontroller>*), this message is printed, and dynamic modification is aborted.

Action:

> Make sure that the number of parent controllers used by an application is less than the number specified as part of the `MaxControllers` attribute, or modify `MaxControllers` to increase the number.

● (ADM, 82) 62. Dynamic modification failed: cannot delete *type* *<object>* unless its state is one of Unknown, Wait, Offline or Faulted.

> This message may appear in the switchlog if there is an attempt to delete a `SysNode` or `SatNode` from a running configuration if the node is not in one of the states Unknown, Offline, Wait or Faulted.

Action:

> Shut down RMS on that host and then do the deletion.

● (ADM, 83) 63. Dynamic modification failed: cannot delete SysNode *<sysnode>* since this RMS monitor is running on this SysNode.

> During dynamic modification the local `SysNode` *<sysnode>* was going to be deleted.

Action:

> Make sure dynamic modification does not contain 'delete *sysnode*;' where *sysnode* is the name of the local node.

● `(ADM, 84) 64. Dynamic modification failed: cannot add`
  `SysNode` *<sysnode>* `since its name is not valid.`

   This message appears in the switchlog if the name *<sysnode>* specified
   as part of the dynamic modification is not resolvable to any known host
   name.

   Action:
   Specify a host name that is resolvable to a network address.

● `(ADM, 85) 65. Dynamic modification failed: timeout expired,`
  `timeout symbol is` *<symbol>*`.`

   If the dynamic modification takes too much time, this message is printed.

   Action:
   Make sure that the network connection between the hosts is functional,
   and also verify that the scripts from newly added resources do not take
   too much time to execute, or that dynamic modification does not add too
   many new nodes, or that the modification file is too big or too complex.

● `(ADM, 86) 66. Dynamic modification failed: application`
  <userapplication> `cannot be deleted since it is controlled by`
  `the controller` *<controller>*`.`

   A controlled application *<userapplication>* cannot be deleted while its
   controller *<controller>* retains the application's name in its `Resource`
   attribute.

   Action:
   Remove the name of the deleted application from the controller's
   `Resource` attribute, or add a new application with the same name, or
   delete the controller together with its controlled application, or change
   the controller's `NullDetector` attribute to 1.

● `(ADM, 87) 67. Dynamic modification failed: only local`
  `attributes such as ScriptTimeout, DetectorStartScript,`
  `NullDetector or MonitorOnly can be modified during local`
  `modification (hvmod -l).`

   The reason for this message is that only the modification of local
   attributes is allowed during local modification.

   Action:
   Make a non-local modification, or modify different attributes.

● (ADM, 88) 68. Dynamic modification failed: attribute
  <attribute> is modified more than once for object *<object>*.

  This message may appear because an attribute of a particular object can
  be modified only once in the same modification file, but *<attribute>* has
  been modified more than once for *<%object>*.

  Action:
  Modify the attribute only once per object.

● (ADM, 89) 69. Dynamic modification failed: cannot rename
  existing object *<sysnode>* to *<othersysnode>* because either there
  is no object named *<sysnode>*, or another object with the name
  <othersysnode> already exists, or a new object with that name
  is being added, or the object is not a resource, or it is a
  SysNode, or it is a controlled application which state will
  not be compatible with its controller.

  This message appears when we try to rename an existing object
  <sysnode> to other node *<othersysnode>* but one of the following condi-
  tions was encountered:
  <othersysnode> is not a valid name.
  <othersysnode> is already used by some other host in the cluster.
  <othersysnode> is not a resource.
  <othersysnode> is a controlled application.

  Action:
  Choose another valid host name.

● (ADM, 90) 70. Dynamic modification failed: cannot change
  attribute Resource of the controller object *<controllernode>*
  from *<oldresource>* to *<newresource>* because some of *<oldresource>*
  are going to be deleted.

  This message appears when the user tries to rename a resource that is
  controlled by a controller object and is going to be deleted.

  Action:
  Make sure deleted applications are not referred to from any controller.

- (ADM, 91) 71. Dynamic modification failed: controller
  <*controller*> has its Resource attribute set to <*resource*>, but
  application named <*userapplication*> is going to be deleted.

  This message appears when the user tries to control a resource
  <*resource*> with a controller <*controller*> but the application associated
  with that resource is going to be deleted.

  Action:
  Make sure the controller's Resource attribute does not refer to a deleted
  application.

- (ADM, 95) Cannot retrieve information about command line
  used when starting RMS. Start on remote host must be
  skipped. Please start RMS manually on remote hosts.

  RMS was started with the −a option but due to some internal error RMS
  could not be started on the remote host. Critical error.

  Action:
  Contact field support. For temporary workaround, try again or start RMS
  manually on each host.

- (ADM, 96) Remote startup of RMS failed <*startupcommand*>.
  Reason: *errorreason*.

  When RMS cannot be started on remote hosts because the command
  <*startupcommand*> failed.

  Action:
  This may occur when some of the hosts are not reachable or the network
  is down.

- (ADM, 98) 72. Dynamic modification failed: controller
  <*controller*> has its Resource attribute set to <*resource*>, but
  some of the controlled applications from this list do not
  exist.

  This message appears when the controller node was not able to find the
  applications controlled by it with the applications running on the host.

  Action:
  Correct your modification file so that the controllers refer only to the
  existing application.

- (ADM, 99) 73. `Dynamic modification failed: cannot change attribute Resource of the controller object` *`<controller>`* `from` *`<oldresource>`* `to` *`<newresource>`* `because one or more of the applications listed in` *`<newresource>`* `is not an existing application or its state is incompatible with the state of the controller, or because the list contains duplicate elements.`

  This message appears when the user tries to change the Resource attribute of the controller object *<controller>* from *<oldresource>* to *<newresource>* because one or more of the applications listed in *<newresource>* is not an existing application or its state is incompatible with the state of the controller, or because the list contains duplicate elements.

  Action:
  Make sure that the applications listed in the resource *<newresource>* are not written more than once or invalid.

- (ADM, 100) 74. `Dynamic modification failed: because a controller` *`<controller>`* `has AutoRecover set to 1, its controlled application` *`<userapplication>`* `cannot have PreserveState set to 0 or AutoSwitchOver set to 1.`

  If an application needs to be controlled by a controller then the applications' attributes PreserveState and AutoSwitchOver need to be 1 and 0 respectively if the controller has its AutoRecover set to 1.

  Action:
  Check the PreserveState and AutoSwitchOver attribute of the application.

- (ADM, 106) `The total number of SysNodes specified in the configuration for this cluster is` *hosts*`. This exceeds the maximum allowable number of SysNodes in a cluster which is` *maxhosts*`.`

  The total number of SysNode objects in the cluster has exceeded the maximum allowable limit.

  Action:
  Make sure that the total number of SysNode objects in the cluster does not exceed *maxhosts*.

- (ADM, 107) The cumulative length of the SysNode names specified in the configuration for the userApplication <*userapplication*> is *length*. This exceeds the maximum allowable length which is *maxlength*.

  The cumulative length of the SysNode names specified in the configuration for application *userapplication* exceeds the maximum allowable limit.

  Action:
  Limit the length of the SysNode names so that they fit within the maximum allowable limit.

- (ADM, 118) 106. Dynamic modification failed: cannot add SatNode <*satnode*> since its rKind <*rkind*> is not consistent with the rKind of the other SatNodes.

  The rKind of SatNode *satnode* is not the same as the rKind of the other SatNodes.

  Action:
  Make sure that the rKind of all the SatNodes is the same.

- (ADM, 125) 107. Dynamic modification failed: The <*attr*> entry <*value*> for SysNode <*sysnode*> matches the <*attr*> entry or the SysNode name for another SysNode.

  The entry *attr* must be unique.

  Action:
  Ensure that the *attr* entry is unique.

3.3   BAS: Startup and configuration errors

- (BAS, 2) Duplicate line in hvgdstartup.

  If RMS detects that a line has been duplicated in the hvgdstartup, it prints this error message. The end result of this is that RMS will exit with exit code 23.

  Action:
  Only unique lines are allowed in hvgdstartup. Remove all the duplicate entries.

- (BAS, 3) `No kind specified in hvgdstartup.`

  In the `hvgdstartup` file, the entry for the detector is not of the form 'gN −t*n* −k*n*', or the −k*n*& option is missing. Since RMS is unable to start, it exits with exit code 23.

  Action:
  Modify the entry for the detector so that the kind (−k*n* option) for the detector is specified properly.

- (BAS, 6) `DetectorStartScript for kind` *<kind>* `cannot be redefined while detector is running.`

  During dynamic modification, there was an attempt to redefine the kind for the `DetectorStartScript`.

  Action:
  Do not attempt to redefine the `DetectorStartScript` when the detector is already running.

- (BAS, 9) `ERROR IN CONFIGURATION FILE:` *message*.

  The *message* can be any one of the following:
  - `Check for SanityCheckErrorPrint`
  - `Object` *<object>* `cannot have its HostName attribute set since it is not a child of any userApplication. Only the direct descendants of userApplication can have the HostName attribute set.`
  - `In basic.C:parentsCount(...)`
  - `The node` *<node>* `belongs to more than one userAppli-cation,` *app1* `and` *app2*. `Nodes must be children of one and only one userApplication node.`
  - `The node` *<node>* `is a leaf node and this type` *<type>* `does not have a detector. Leaf nodes must have detectors.`
  - `The node` *<node>* `has an empty DeviceName attribute. This node uses a detector and therefore it needs a valid DeviceName attribute.`
  - `The rName is` *<rname>*, `its length` *length* `is larger than max length` *maxlength*.
  - `The DuplicateLineInHvgdstartup is` *<number>*, `so the hvgdstartup file has a duplicate line.`
  - `The NoKindSpecifiedForGdet is` *<number>*, `so no kind specified in hvgdstartup.`
  - `Failed to load a detector of kind` *<kind>*.

- The node *<node>* has an invalid rKind attribute. Nodes
  of type gResource must have a valid rKind attribute.
- The node *<node>* has a ScriptTimeout value that is less
  than its detector report time. This will cause a script
  timeout error to be reported before the detector can
  report the state of the resource. Increase the **Script-
  Timeout** value for *objectname* (currently *value* seconds) to
  be greater than the detector cycle time (currently *value*
  seconds).
- Node *<node>* has no detector while all its children's
  "MonitorOnly" attributes are set to 1.
- The node *<node>* has both attributes "LieOffline" and
  "ClusterExclusive" set. These attributes are incom-
  patible; only one of them may be used.
- The type of object *<object>* cannot be or and and at the
  same time.
- Object *<object>* is of type and, its state is online, but
  not all children are online.

Action:
> Verify the above description and change the configuration appropriately.

- (BAS, 14) ERROR IN CONFIGURATION FILE: The object *<object>*
  belongs to more than one userApplication, *userapplication1* and
  userapplication2. Objects must be children of one and only one
  userApplication object.

  > An object was encountered as a part of more than one user applications.
  > RMS applications cannot have common objects.

Action:
> Redesign your configuration so that no two applications have common
> objects.

- (BAS, 15) ERROR IN CONFIGURATION FILE: The object *<object>* is
  a leaf object and this type *<type>* does not have a detector.
  Leaf objects must have detectors.

  > An object that has no children objects (i.e. a leaf object) is of type *type*
  > that has no detectors in RMS. All leaf objects in RMS configurations must
  > have detectors.

Action:
> Redesign your configuration so that all leaf objects have detectors.

- (BAS, 16) ERROR IN CONFIGURATION FILE: The object *object* has an empty DeviceName attribute. This object uses a detector and therefore it needs a valid DeviceName attribute.

    Critical internal error. If this message appears in switchlog, it indicates a severe problem in the base monitor.

    Action:
    Contact field support.

- (BAS, 17) ERROR IN CONFIGURATION FILE: The rName is *<rname>*, its length *length* is larger than max length *maxlength*.

    The value of the rName attribute exceeds the maximum length of *maxlength* characters.

    Action:
    Specify shorter rName.

- (BAS, 18) ERROR IN CONFIGURATION FILE: The duplicate line number is *<linenumber>*.

    this message prints out a line number of the duplicate line in hvgdstartup file.

    Action:
    Make sure that file hvgdstartup has no duplicate lines.

- (BAS, 19) ERROR IN CONFIGURATION FILE: The NoKindSpecified-ForGdet is *<kind>*, so no kind specified in hvgdstartup.

    The kind has not been specified for the generic detector in the hvgdstartup file.

    Action:
    Specify the kind for the generic detector in hvgdstartup.

- (BAS, 23) ERROR IN CONFIGURATION FILE: DetectorStartScript for object *object* is not defined. Objects of type *type* should have a valid DetectorStartScript attribute.

    Object *object* does not have its DetectorStartScript defined.

    Action:
    Make sure that the DetectorStartScript is defined for object *object*.

- (BAS, 24) ERROR IN CONFIGURATION FILE: The object *object* has an invalid rKind attribute. Objects of type gResource must have a valid rKind attribute.

  Object *object* has an invalid rKind attribute.

  Action:
  Make sure that the object *object* has a valid rKind attribute.

- (BAS, 25) ERROR IN CONFIGURATION FILE: The object *object* has a ScriptTimeout value that is less than its detector report time. This will cause a script timeout error to be reported before the detector can report the state of the resource. Increase the ScriptTimeout value for *object* (currently *seconds* seconds) to be greater than the detector cycle time (currently *detectorcycletime* seconds).

  The ScriptTimeout value is less than the detector cycle time. This will cause the resource to appear faulted when being brought Online or Offline.

  Action:
  Make the value of ScriptTimeout greater than the detector report time.

- (BAS, 26) ERROR IN CONFIGURATION FILE: The type of object <*object*> cannot be 'or' and 'and' at the same time.

  Each RMS object must be of a type derived from or or and types, but not both. If this message appears in the switchlog, it indicates of a severe corruption of the RMS executable.

  Action:
  Contact field support.

- (BAS, 27) ERROR IN CONFIGURATION FILE: object <*object*> is of type 'and', its state is online, but not all children are online.

  This message may appear during dynamic modification, when the existing configuration is checked before applying the modification. If this message appears, the dynamic modification will not proceed.

  Action:
  Make sure that online objects of type and have all their children in online states, only then apply dynamic modification.

- (BAS, 29) ERROR IN CONFIGURATION FILE: object *<object>* cannot have its HostName attribute set since it is not a child of any userApplication.

  An object that is not a child of a userApplication has its HostName attribute set. Only children of the userApplication object can and must have its HostName attribute set.

  Action:
  Eliminate the HostName attribute from the definition of the object, or disconnect the userApplication object from this object, making this object a child of another, non-userApplication object.

- (BAS, 30) ERROR IN CONFIGURATION FILE: The object *object* has both attributes "LieOffline" and "ClusterExclusive" set. These attributes are incompatible; only one of them may be used.

  Both attributes LieOffline and ClusterExclusive are set for the same RMS object. Only one of them can be set for the same object.

  Action:
  Eliminate one or both settings from the RMS object *object*.

- (BAS, 31) ERROR IN CONFIGURATION FILE: Failed to load a detector of kind *<kind>*.

  A detector was not able to be started by the RMS base monitor.

  Action:
  Make sure detector executable is present in the right place and has executable privileges.

- (BAS, 32) ERROR IN CONFIGURATION FILE: Object *<object>* has no detector while all its children's <MonitorOnly> attributes are set to 1.

  An object without a detector has all its children's MonitorOnly attributes set to 1. An object without a detector must have at least one child for which MonitorOnly is set to 0.

  Action:
  Change the configuration so that each object without a detector has at least one child with its MonitorOnly set to 0.

- (BAS, 36) ERROR IN CONFIGURATION FILE: The object *object* has both attributes "MonitorOnly" and "ClusterExclusive" set. These attributes are incompatible; only one of them may be used.

   Both attributes `MonitorOnly` and `ClusterExclusive` are set for the same RMS object. Only one of them can be set for the same object.

   Action:
   Eliminate one or both settings from the RMS object *object*.

- (BAS, 37) ERROR IN CONFIGURATION FILE: The satApplication <*satapp*> has no children, or its children are not valid resources.

   The satApplication *satapp* has no children or they are not valid resources.

   Action:
   Ensure that *satapp* has valid resources as its children.

- (BAS, 38) ERROR IN CONFIGURATION FILE: The satService <*satserv*> has no children, or its children are not valid resources.

   The satService *satserv* has no children or they are not valid resources.

   Action:
   Ensure that *satserv* has valid resources as its children.

- (BAS, 39) ERROR IN CONFIGURATION FILE: The gResource <*gresource*> belongs to a satellite configuration and it is not allowed to have any children.

   The gResource *gresource* may not have any children.

   Action:
   Ensure that *gresource* does not have any children.

- (BAS, 40) ERROR IN CONFIGURATION FILE: The object <*object*> belongs to more than one satApplication, *satapplication1* and *satapplication2*. Objects must be children of one and only one satApplication object.

   An object was encountered as a part of more than one satApplication. RMS satApplications cannot have common objects.

Action:

> Redesign your configuration so that no two applications have common objects.

● (BAS, 41) ERROR IN CONFIGURATION FILE: The object *<object>* belongs to more than one satService, *satservice1* and *satservice2*. Objects must be children of one and only one satService object.

> An object was encountered as a part of more than one satService. RMS satServices cannot have common objects.

Action:

> Redesign your configuration so that no two satServices have common objects.

● (BAS, 42) ERROR IN CONFIGURATION FILE: The gResource *<gresource>* belonging to a satellite configuration does not have its HostName attribute set. Please ensure that the HostName attribute is set for this gResource.

> The gResource *gresource* does not have its HostName attribute set.

Action:

> Ensure that the HostName attribute is set for the gResources belonging to a satellite configuration.

3.4   BM: Base monitor

● (BM, 8) Failed sending message *<message>* to object *<object>* on host *<host>*.

> When RMS encounters some problems in transmitting the message *<message>* to some other host in the cluster, it prints this message. This could be due to the fact that the RMS on the other host is down or there might be a network problem.

Action:

> Make sure that the RMS is running on the other hosts in the cluster and also whether there are any network issues.

● (BM, 13) S4: no symbol for object *<object>* in .inp file, line = *linenumber*.

  RMS internal error.

  Action:
  Contact field support.

● (BM, 14) S6: local queue is empty on read directive in line: *linenumber*.

  RMS internal error.

  Action:
  Contact field support.

● (BM, 15) S2: destination object *<object>* is absent in line: *linenumber*.

  RMS internal error.

  Action:
  Contact field support.

● (BM, 16) S2: sender object *<object>* is absent in line: *linenumber*.

  RMS internal error.

  Action:
  Contact field support.

● (BM, 17) 53. Dynamic modification failed: line *linenumber*, cannot build an object of unknown type *<symbol>*.

  An object of unknown type is added during dynamic modification.

  Action:
  Use only objects of known types in configuration files.

● (BM, 18) 54. Dynamic modification failed: line *linenumber*, cannot set value for attribute *<attribute>* since object *<object>* does not exist.

  An attribute of a non-existing object cannot be modified.

Action:

Modify attributes only for existing objects.

- (BM, 19) 39. Dynamic modification failed: line *linenumber*, cannot modify attribute *<attribute>* of object *<object>* with value *<value>*.

Invalid attribute is specified for modification.

Action:

Modify only valid attributes.

- (BM, 20) 77. Dynamic modification failed: line *linenumber*, cannot build object *<object>* because its type *<symbol>* is not a user type.

An object *<object>* of a system type *<symbol>* is specified during dynamic modification.

Action:

Use only valid resource types when adding new objects to configuration.

- (BM, 21) 78. Dynamic modification failed: cannot delete object *<object>* because its type *<symbol>* is not a user type.

An object *<object>* of a system type *<symbol>* is specified for deletion.

Action:

Delete only objects that are valid resource types.

- (BM, 23) 80. Dynamic modification failed: The <Follow> attribute for controller *<controller>* is set to 1, but the content of a PriorityList of the controlled application *<controlleduserapplication>* is different from the content of the PriorityList of the application *<userapplication>* to which *<controller>* belongs.

This message appears when the PriorityList of the controlled application *<controlleduserapplication>* is different from the content of the PriorityList of the application *<userapplication>* to which the controller *<controller>* belongs.

Action:

Make sure that the PriorityList of the controller and the controlled application is same.

- (BM, 24) 81. Dynamic modification failed: some resource(s) supposed to come standby failed.

  During dynamic modification, an attempt was made to add new resource(s) to a resource that was in Standby mode, but the resources could not also be brought into Standby mode.

  Action:
  Analyze your configuration to make sure that standby capable resources can get to the Standby state.

- (BM, 25) 82. Dynamic modification failed: standby capable controller *<controller>* cannot control application *<userapplication>* which has no standby capable resources on host *<sysnode>*.

  In order for an application *<userapplication>* to be controlled by a controller *<controller>* the application *<userapplication>* has to have at least one standby capable resource on host *<sysnode>*.

  Action:
  Make sure that the controlled application has at least one standby capable controller or make sure that the controllers are not standby capable.

- (BM, 26) 83. Dynamic modification failed: controller *<controller>* cannot have attributes StandbyCapable and Ignore-StandbyRequest both set to 0.

  This message appears when user sets both controller attributes Stand-byCapable **and** IgnoreStandbyRequest **to 1.**

  Action:
  Make sure that only one is set to 1 and other to 0.

- (BM, 29) 84. Dynamic modification failed: controller object *<controller>* cannot have its attribute 'Follow' set to 1 while one of OnlineTimeout or StandbyTimeout is not null.

  The controller node *<controller>* should have one of its attributes Online-Timeout or StandbyTimeout be null to allow the attribute Follow to be 1.

  Action:
  Set the attributes accordingly and try again.

- (BM, 42) 87. Dynamic modification failed: application `<`*userapplication*`>` is not controlled by any controller, but has one of its attributes ControlledSwitch or Controlled- Shutdown set to 1.

  This message appears when the user wants the application *<userapplication>* to be controlled by a controller but one or more of the applications' attributes ControlledSwitch or ControlledShutdown is set to 1.

  Action:
  Set the attributes accordingly and try again.

- (BM, 46) 89. Dynamic modification failed: cannot modify a global attribute `<`*attribute*`>` locally on host `<`*hostname*`>`.

  The user cannot modify global attributes *<attribute>* like Detec- torStartScript or NullDetector or NonCritical locally on a host *<hostname>*.

  Action:
  Modify the *attribute* globally or modify locally a different attribute.

- (BM, 54) The RMS-CF-CIP mapping cannot be determined for any host due to the CIP configuration file `<`*configfilename*`>` missing entries. Please verify all entries in `<`*configfilename*`>` are correct and that CF and CIP are fully configured.

  CIP configuration file has missing entries.

  Action:
  Make sure that the CIP configuration has entries for all the RMS hosts that are running in a cluster.

- (BM, 59) Error *errno* while reading line `<`*linenumber*`>` of .dob file -- `<`*errorreason*`>`.

  During dynamic modification, the base monitor reads its configuration from a '.dob' file. When this file cannot be read, this message appears in the switchlog. The specific OS error is indicated in *errno* and *errorreason*.

  Action:
  Make sure the host conditions are such that .dob file can be read without errors.

- (BM, 68) Cannot get message queue parameters using sysdef, errno = *<errno>*, reason: *<reason>*.

     While obtaining message queue parameters, sysdef was not able to communicate them back to the base monitor. The values of *errno* and *reason* indicate the kind of error.

     Action:
       Contact field support.

- (BM, 71) Dynamic modification failed: Controller *<controller>* has its attribute Follow set to 1. Therefore, its attribute IndependentSwitch must be set to 0, and its controlled application *<application>* must have attributes AutoSwitchOver == "No" StandbyTransitions="No" AutoStartUp=0 Controlled-Switch = 1 ControlledShutdown = 1 PartialCluster = 0. However, the real values are IndependentSwitch = *<isw>* AutoSwitchOver = *<asw>* StandbyTransitions = *<str>* AutoStartUp = *<asu>* ControlledSwitch = *<csw>* ControlledShutdown = *<css>* PartialCluster = *<pcl>*.

     When the controller's Follow attribute is set, other attributes such as IndependentSwitchover, AutoSwitchOver, StandbyTransitions, AutoStartUp, ControlledSwitch, ControlledShutdown and PartialCluster must have the values 0, No, No, 0, 1, 1 and 0 respectively. However, this condition is violated in the configuration file.

     Action:
       Supply a valid combination of attributes for the controller and its controlled user application.

- (BM, 72) 91. Dynamic modification failed: Controller *<controller>* with the <Follow> attribute set to 1 belongs to an application *<application>* which PersistentFault is *<appfault>*, while its controlled application *<controlledapplication>* has its PersistentFault *<_fault>*.

     If controller has its Follow set to 1 then all its controlled applications must have the same value for the attribute PersistentFault as the parent application of the controller.

     Action:
       Check and correct the configuration.

● (BM, 73) The RMS-CF interface is inconsistent and will require operator intervention. The routine "*routine*" failed with error code *errorcode* - "*errorreason*".

> This is a generic message indicating that the execution of the routine *routine* failed due to the reason *errorreason* and hence the RMS-CF interface is inconsistent. Depending on which routine *routine* has failed, the base monitor can exit with any one of the exit codes 132, 133, 134, 135, 136, 137, 138 or 95.

Action:
> Contact field support.

● (BM, 74) The attribute DetectorStartScript and hvgdstartup file cannot be used together. The hvgdstartup file is for backward compatibility only and support for it may be withdrawn in future releases. Therefore it is recommended that only the attribute DetectorStartScript be used for setting new configurations.

> The attribute DetectorStartScript and the file hvgdstartup are mutually exclusive.

Action:
> Make sure that the DetectorStartScript be used for setting new configurations as support for hvgdstartup may be discontinued in future releases.

● (BM, 75) 88. Dynamic modification failed: controller <*controller*> has its attributes SplitRequest, IgnoreOnlineRe-quest, and IgnoreOfflineRequest set to 1. If SplitRequest is set to 1, then at least one of IgnoreOfflineRequest or IgnoreOnlineRequest must be set to 0.

> Invalid combination of controller attributes is encountered. If both IgnoreOfflineRequest and IgnoreOnlineRequest are set to 1, then no request will be propagated to the controlled application(s), so no request can be split.

Action:
> Provide a valid combination of the controller attributes.

- (BM, 80) 92. `Dynamic modification failed: controller` *controller* `belongs to the application` *application* `which` `AutoSwitchOver attribute has "ShutDown" option set, but its` `controlled application` *controlled* `has not.`

  If a controlling application has its `AutoSwitchOver` attribute set with the option "Shutdown", then all applications controlled by the controllers that belong to this controlling application must also have their `AutoSwitchOver` attributes having the option "Shutdown" set as well.

  Action:
  Provide correct settings for the `AutoSwitchOver` attributes.

- (BM, 81) 93. `Dynamic modification failed: local controller` `attributes such as NullDetector or MonitorOnly cannot be` `modified during local modification (hvmod -l).`

  The reason for this message is that the modification of local controller attributes such as `NullDetector` or `MonitorOnly` are allowed only during global modification.

  Action:
  Make a non-local modification, or modify different attributes.

- (BM, 90) 94. `Dynamic modification failed: The length of` `object name` *object* `is` *length*. `This is greater than the maximum` `allowable length name of` *maxlength*.

  The length of object name is greater than the maximum allowable length.

  Action:
  Ensure that the length of the object name is smaller than *maxlength*.

- (BM, 92) 95. `Dynamic modification failed: a non-empty value` *value* `is set to <ApplicationSequence> attribute of a non-` `scalable controller` *controller*.

  A non-scalable controller cannot have its `ApplicationSequence` attribute set to a non-empty value.

  Action:
  Provide correct settings for the `ApplicationSequence` and `Scalable` attributes.

- (BM, 94) 97. Dynamic modification failed: the Applicationt-Sequence attribute of a scalable controller *<controller>* includes application name *<hostname>*, but this name is absent from the list of controlled applications set to the value of *<resouce>* in the attribute <Resource>.

    The ApplicationSequence attribute of a scalable controller includes an application name absent from the list of the controlled applications.

    Action:
    Provide correct settings for ApplicationSequence and Resource attributes of the controller.

- (BM, 96) 94. Dynamic modification failed: a scalable controller *<controller>* has its attributes <Follow> set to 1 or <IndependentSwitch> set to 0.

    A scalable controller must have its attribute Follow set to 0 and <IndependentSwitch> set to 1.

    Action:
    Provide correct settings for the Follow, IndependentSwitch, and Scalable attributes.

- (BM, 97) 95. Dynamic modification failed: controller *<controller>* attribute <ApplicationSequence> is set to *<applicationsequence>* which refers to application(s) not present in the configuration.

    A scalable controller must list only existing applications in its ApplicationSequence attribute.

    Action:
    Provide correct settings for attribute ApplicationSequence.

- (BM, 98) 96. Dynamic modification failed: two scalable controllers *<controller1>* and *<controller2>* control the same application *<application>*.

    Only one scalable controller can control an application.

    Action:
    Fix RMS configuration.

● (BM, 99) 97. Dynamic modification failed: controlled appli-
cation *<controlledapp>* runs on host *<hostname>*, but it is
controlled by a scalable controller *<scontroller>* which belongs
to an application *<controllingapp>* that does not run on that
host.

> Hostname mismatch between controlled and controlling applications.
> Controlling application must run on all the hosts where the controlled
> applications are running.

Action:
> Fix RMS configuration.

● (BM, 101) 99. Dynamic modification failed: controlled appli-
cation *<controlledapp>* runs on host *<hostname>*, but it is
controlled by a scalable controller *<scontroller>* which belongs
to a controlling application *<controllingapp>* that does not
allow for the controller to run on that host.

> Hostname mismatch between controlled and controlling applications.
> Controlling application must run on all the hosts where the controlled
> applications are running.

Action:
> Fix RMS configuration.

● (BM, 103) Dynamic modification failed: Controller *<controller>*
has its attribute Follow set to 1 and the controlled appli-
cation *<application>* has StandbyCapable resources. Therefore
the controller itself must have StandbyCapable set to 1 and
IgnoreStandbyRequest must be set to 0.

> When the controller's Follow attribute is set and the controlled appli-
> cation has StandbyCapable resources, the controller must have
> StandbyCapable set and IgnoreStandbyRequest must be disabled.
> Otherwise Standby requests will not properly been propagated to the
> controlled application.

Action:
> Supply a valid combination of attributes for the controller and its
> controlled user application.

● (BM, 105) 100. Dynamic modification failed: Invalid kind of generic resource specified in DetectorStartScript *<script>* for object *<object>*.

Wrong value is supplied for a flag −k in the detector startup script.

Action:
Fix RMS configuration.

● (BM, 106) The rKind attribute of object *<object>* does not match the value of the '−k' flag of its associated detector.

Values for rKind attribute and flag −k of the detector startup line do not match.

Action:
Fix RMS configuration.

● (BM, 107) Illegal different values for rKind attribute in object *<object>*.

Different values for rKind attribute are encountered within the same object.

Action:
Fix RMS configuration.

● (BM, 108) 101. Dynamic modification failed: Scalable controller *<object>* cannot have its attribute <SplitRequest> set to 1.

Setting controller attributes Scalable and SplitRequest is mutually exclusive.

Action:
Fix RMS configuration.

- (BM, 109) 102. Dynamic modification failed: Application <*application*> has its attribute PartialCluster set to 1 or is controlled, directly or indirectly, via a Follow controller that belongs to another application that has its attribute PartialCluster set to 1 -- this application <*application*> cannot have a cluster exclusive resource <*resource*>.

    An exclusive resource cannot belong to an application with the attribute PartialCluster set to 1, or cannot be controlled, directly or indirectly, by a Follow controller from an application with the attribute PartialCluster set to 1.

    Action:
    Fix RMS configuration.

- (BM, 110) 103. Dynamic modification failed: Application <*application*> is controlled by a scalable controller <*controller*>, therefore it cannot have its attribute <ControlledShutdown> set to 1 while its attribute <AutoSwitchOver> includes option <ShutDown>.

    An application controlled by a scalable controller cannot have ControlledShutdown set to 1 and AutoSwitchOver including the option <ShutDown> at the same time.

    Action:
    Correct RMS configuration.

- (BM, 111) 104. Dynamic modification failed: Line #*line* is too big.

    A line in a configuration file is too big.

    Action:
    Fix RMS configuration, so that each line takes less than 2000 bytes.

- (BM, 114) 107. Dynamic modification failed: The SatNode <*satnode*> specified is already an existing SysNode entry.

    The SatNode *satnode* is a duplicate of an existing SysNode.

    Action:
    Change the name of the SatNode and retry the modification.

3.5 CML: Command line

● (CML, 11) Option (*option*) requires an operand.

> Certain options for hvcm require an argument. If hvcm has been invoked without the argument, this message appears along with the usage and RMS exits with exit code 3.

> *Action:*
> Check the hvcm man page for correct usage.

● (CML, 12) Unrecognized option *option*.

> The option provided is not a valid one.

> *Action:*
> Check the hvcm man page for correct usage.

● (CML, 17) Incorrect range argument with -l option.

> The number for the -l option is not correct. Check the range.

> *Action:*
> Check the man page for hvcm for range argument with -l option.

● (CML, 18) Log level *<loglevel>* is too large. The valid range is 1..*maxloglevel* with the -l option.

> If the loglevel *loglevel* specified with -l option for hvcm is greater than the maximum possible loglevel *maxloglevel*, this message is printed, and RMS exits with exit code 4.

> *Action:*
> Specify a loglevel between 1 and *maxloglevel* for 'hvcm -l'.

● (CML, 19) Invalid range *<low – high>*. Within the '-l' option, the end range value must be larger than the first one.

> When a range of loglevels has been specified with -l option for hvcm, if the value of the end range *high* is smaller than the value of *low*, this message appears and RMS exits with exit code 4.

> *Action:*
> Specify the end range value to be higher than the initial end range value.

● (CML, 20) `Log level must be numeric.`

> If the log level specified with the −l option for `hvcm` is not a number, this message is printed, and RMS exits with exit code 4.

Action:
> Specify a numeric value for the log level.

● (CML, 21) `0 is an invalid range value. 0 implies all values` `If a range is desired, the valid range is 1..`*maxloglevel* `with` `the −l option.`

> If the log level specified with the −l option of `hvcm` is outside the valid range, this message is printed and RMS exits with exit code 4.

Action:
> The valid range for the −l option of `hvcm` is 1..*maxloglevel*.

3.6 CRT: Contracts and contract jobs

● (CRT, 1) `FindNextHost: local host not found in priority list` `of` *nodename*.

> The RMS base monitor maintains a priority list of all the hosts in the cluster. Under normal circumstances, the local host should always be present in the list. If this is not the case, this message is printed.

Action:
> Contact field support.

● (CRT, 2) `cannot obtain the NET_SEND_Q queue.`

> RMS uses internal queues for sending contracts. Contracts are messages that are transmitted between the hosts in a cluster that ensure the hosts are synchronized with respect to a particular operation. The messages may be transmitted between processes on the same host or processes on different hosts. If there is a problem with the queue `NET_SEND_Q` that is being used to transmit these contracts, this message is printed in the switchlog.

Action:
> Contact field support.

- (CRT, 3) `Message send failed.`

    When RMS tries to send a message to another host in the cluster, if the delivery of this message over the queue NET_SEND_Q has failed, this message is printed. This could be due to the fact that the host that is to receive the message has gone down or there is a problem with the cluster interconnect.

    Action:

    Check to make sure that the other hosts in the cluster are all alive and make sure that none of them are experiencing any network problems.

- (CRT, 4) *object*: *type* `Contract retransmit failed: Message Id =` *messageid* `see bmlog for contract details.`

    When RMS on one host sends a contract over the queue NET_SEND_Q to another host (or itself, if there is only one host in the cluster), it tries to transmit this contract a certain number of times that is determined internally. If the message transmission fails after all attempts, this message is printed to the switchlog and the contract is discarded. (Note: UAP contracts are not discarded.)

    Action:

    Make sure that there is no problem with the cluster interconnect. If contract retransmissions occur for userApplication contracts, make sure that the cluster is in consistent condition (i.e., no userApplication is online on more than one host, no SysNode is in a pending wait state, etc.).

- (CRT, 5) `The contract` *<crtname>* `is being dropped because the local host` *<crthost>* `has found the host originator` *<otherhost>* `in state` *<state>*. `That host is expected to be in state Online. Please check the interhost communication channels and make sure that these hosts see each other Online.`

    The local host *crthost* sees the contract host originator in state *state* when it is expected to be in state Online.

    Action:

    Make sure that the interhost communication channels are working correctly and that the hosts see each other online.

3.7 CTL: Controllers

- (CTL, 1) Controller *<controller>* will not operate properly since its controlled resource *<resource>* is not in the config-uration.

    This message appears when a resource is not in the configuration that is controlled by a controller and the controller's `NullDetector` attribute is set to `off`.

    Action:
    The controlled resource must be present in the configuration for the controller to work properly.

- (CTL, 2) Controller *<controller>* detected more than one controlled application Online. This has lead to the controller fault. Therefore, all the online controlled application will now be switched offline.

    If the controller *controller* has two or more of the controlled applications Online on one or more hosts, then the controller faults.

    Action:
    Make sure that more than one controlled application for a controller is not Online.

3.8 CUP: userApplication contracts

- (CUP, 2) *object*: cluster is in inconsistent condition current online host conflict, received: *host*, local: *onlinenode*.

    The cluster hosts were unable to determine which host is responsible for a particular `userApplication`. The most likely reason for this is an erroneous system administrator intervention (e.g., a forced `hvswitch` request) that left the `userApplication` Online on more than one host simultaneously.

    Action:
    Analyze the cluster inconsistency and perform the appropriate action to resolve it. If the application is online on more than one host, shut down ('hvutil -f') the `userApplication` on all but one host.

- (CUP, 3) *object* is already waiting for an event cannot set timer!

  Critical internal error.

  Action:
  Contact field support.

- (CUP, 5) *object* received unknown contract.

  The contract received by the node from the application is not recognizable. Critical internal error.

  Action:
  Contact field support.

- (CUP, 7) *userapplication* is locally online, but is also online on another host.

  The user application is already online on other host and is also online in current host.

  Action:
  User application can only be online on one host. Make sure the application is offline on all but one of the hosts. If this is not the case use 'hvutil -f' to bring the userApplication to an Offline state on the superfluous hosts.

- (CUP, 8) *object*: could not get an agreement about the current online host; cluster may be in an inconsistent condition!

  The cluster hosts were unable to determine which host is responsible for a particular userApplication. The most likely reason for this is an erroneous system administrator intervention (e.g., a forced hvswitch request) that left the userApplication Online on more than one host simultaneously.
  Note: This message corresponds to (CUP, 2). While (CUP, 8) is printed on the contract originator, (CUP, 2) is printed on the non-originator hosts.

  Action:
  Analyze the cluster inconsistency and perform the appropriate action to resolve it. If the application is online on more than one host, shut down ('hvutil -f') the userApplication on all but one host.

3.9　DET: Detectors

● (DET, 1) FAULT REASON: Resource *<resource>* transitioned to a Faulted state due to a child fault.

　　This message appears when the child faulted unexpectedly thereby causing the resource to fault.

　　Action:
　　Check to see why the child resource has faulted and based on this take corrective action.

● (DET, 2) FAULT REASON: Resource *<resource>* transitioned to a Faulted state due to a detector report.

　　This message is printed when the detector unexpectedly reports Faulted state.

　　Action:
　　Check to see why the resource has faulted and take appropriate action.

● (DET, 3) FAULT REASON: Resource *<resource>* transitioned to a Faulted state due to a script failure.

　　This message appears when the detector failed to execute the script for a resource.

　　Action:
　　Ensure that there is nothing wrong with the script and also check the resource for any problems.

● (DET, 4) FAULT REASON: Resource *<resource>* transitioned to a Faulted state due to a FaultScript failure. This is a double fault.

　　When a resource faults due to some reason, it runs its Fault script, but in this case the Fault script failed to execute for that resource.

　　Action:
　　Check to see if there is a problem with the resource or with the Fault script.

- (DET, 5) FAULT REASON: Resource *<resource>* transitioned to a Faulted state due to the resource failing to come Offline after running its OfflineScript (*offlinescript*).

  After a resource executes its offline script, it is expected to come Offline. If it does not change its state, or transitions to a state other than Offline within the period of seconds specified by its `ScriptTimeout` attribute, the resource is considered as being Faulted.

  Action:
  Make sure the Offline script moves the resource into Offline state.

- (DET, 6) FAULT REASON: Resource *<resource>* transitioned to a Faulted state due to the resource failing to come Online after running its OnlineScript (*onlinescript*).

  After a resource executes its online script, it is expected to come Online. If it does not change its state, or transitions to a state other than Online within the period of seconds specified by its `ScriptTimeout` attribute, the resource is considered as being Faulted.

  Action:
  Make sure the Online script moves the resource into Online state.

- (DET, 7) FAULT REASON: Resource *<resource>* transitioned to a Faulted state due to the resource unexpectedly becoming Offline.

  This message appears when the resource becomes Offline unexpectedly.

  Action:
  Check to see why the resource suddenly transitioned to the Offline state.

- (DET, 11) DETECTOR STARTUP FAILED: Corrupted command line *<commandline>*.

  Critical internal error. This message occurs when the command line is empty or has some incorrect value.

  Action:
  Contact field support.

- (DET, 12) DETECTOR STARTUP FAILED *<detector>*. REASON: *error-reason*.

    If the detector *detector* could not be started due to *errorreason*, this message is printed. The reason *errorreason* could be any one of the following:
    – The detector *detector* does not exist.
    – The detector *detector* does not have execute permission.
    – The process for the detector could not be spawned.
    – If the number of processes created by the base monitor at the same time is greater than 128.

    Action:
    Depending on what the reason for the error is take appropriate action.

- (DET, 13) Failed to execute script *<script>*.

    The detector script is not good or the format is not good.

    Action:
    Check the detector startup script.

- (DET, 24) FAULT REASON: Resource *<resource>* transitioned to a Faulted state due to the resource failing to come Standby after running its OnlineScript (*onlinescript*).

    After a resource executes its online script during standby request, it is expected to come Standby. If it does not change its state, or transitions to a state other than Standby or Online within the period of seconds specified by its ScriptTimeout attribute, the resource is considered as being Faulted.

    Action:
    Make sure the Online script moves the resource into Standby or Online state during standby request.

- (DET, 26) FAULT REASON: Resource *<resource>* transitioned to a Faulted state due to the resource failing to come Online.

    This message appears when the resource fails to come Online after executing it Online scripts that may transition the state of the resource to faulted.

    Action:
    Check to see what prevented the resource *resource* from coming Online.

- (DET, 28) *<object>*: `CalculateState() was invoked for a non-local object! This must never happen. Check for possible configuration errors!`

  During the processing of a request within the state engine, a "request or response token" was delivered to an object that is not defined for the local host. Critical internal error.

  Action:
  Contact field support.

- (DET, 33) `DETECTOR STARTUP FAILED: Restart count exceeded.`

  When a detector dies, RMS attempts to restart it. If a detector successfully restarts and once again dies too many times within one minute, RMS assumes there is a problem, terminates the restart cycle, and prints this message.

  Action:
  Contact field support.

- (DET, 34) `No heartbeat has been received from the detector with pid` *<pid>*, *<startupcommand>*, `during the last` *<seconds>* `seconds. The base monitor will send the process a SIGALRM to interrupt the detector if it is currently stalled waiting for the alarm.`

  In order to avoid stalling of RMS detectors, each detector periodically sends a heartbeat message to the base monitor. When the heartbeat is missing for a period of time, the base monitor prints this message into switchlog. The base monitor will send an alarm signal to the stalled process to ensure the detector will properly handle its main loop responsibilities. If the amount of time stated since the last time the base monitor had received the heartbeat from the detector exceeds 300 seconds, then the message may indicate the base monitor is not allowed to run. Currently, the base monitor is a real-time process, but not locked in memory. This message may also occur because the bm process has been swapped out and has not had a chance to run again.

  Action:
  Make sure that the base monitor and detector are active using system tools such as truss(1) or strace(1). If the loss of heartbeat greatly exceeds the 300 second timeout, then this may require that system swap or main memory is insufficient.

3.10   GEN: Generic detector

● (GEN, 1) Usage: *command* -t time_interval -k kind [-d]

   <*command*> has been invoked in a way that does not conform to its
   expected usage.

   Action:
   Use the specified syntax for the command.

● (GEN, 2) Memory lock failed.

   Action:
   Critical error. Contact field support.

● (GEN, 3) Cannot open *command* log file.

   The file <*command*>log used for logging could not be opened.

   Action:
   Contact field support.

● (GEN, 4) failed to create mutex: *directory*

   The various RMS commands like hvdisp, hvswitch, hvutil and
   hvdump utilize the lock files from the directory <*directory*> for signal
   handling purposes. These files are deleted after these commands are
   completed. The locks directory is also cleaned when RMS starts up. If
   they are not cleaned for some reason, this message is printed, and RMS
   exits with exit code 99.

   Action:
   Make sure that the locks directory <*directory*> exists.

● (GEN, 5) *command*: failed to get information about RMS base
   monitor bm!

   The generic detector <*command*> was unable to get any information
   about the base monitor.

   Action:
   Contact field support.

- (GEN, 7) *command*: `failed to lock virtual memory pages, errno` `=` *value*, `reason:` *reason*.

    The generic detector <*command*> was not able to lock its virtual memory pages in physical memory.

    Action:
    Contact field support.

3.11   INI: init script

- (INI, 1) `Cannot open file` *dumpfile*, `errno =` *errno*: *explanation*.

    This message appears when the file <*dumpfile*> failed to open because of the error code <*errno*>, explained in <*explanation*>.

    Action:
    Correct the problem according to <*explanation*>.

- (INI, 9) `Cannot close file` *dumpfile*, `errno =` *errno*: *explanation*.

    This message appears when the file <*dumpfile*> failed to close because of the error code <*errno*>, explained in <*explanation*>.

    Action:
    Correct the problem according to <*explanation*>.

3.12   MAS: Master process

- (MAS, 3) `The operation` *func* `failed with error code` *errorcode*.

    The operation *func* failed with error code *errorcode*.

    Action:
    Contact field support.

- (MAS, 4) `Error becoming a real time process:` *errorreason*

    The master process could not be made into a real time process due to *errorreason*.

    Action:
    Contact field support.

- (MAS, 5) Error setting up real time parameters: *errorreason*

    There was an error *errorreason* setting up the parameters for the master process.

    Action:
    Contact field support.

3.13   MIS: Miscellaneous

- (MIS, 1) No space for object.

    Action:
    Critical error. Contact field support.

- (MIS, 7) The ccd lock file could not be cleaned up.

    In an AS scenario, RMS was unable to clean up the lock file.

    Action:
    Contact field support.

- (MIS, 8) RMS was unable perform a forced shutdown of ccd.

    RMS was unable perform a forced shutdown of ccd. Critical error.

    Action:
    Contact field support.

3.14   QUE: Message queues

- (QUE, 13) RCP fail: *filename* is being copied.

    An attempt was made to copy the file *filename* when there was another copy in progress.

    Action:
    Make sure that concurrent copies of the same file do not occur.

- (QUE, 14) RCP fail: fwrite errno *errno*.

    There was a problem while transferring files from one cluster host to the other.

Action:
> Take action based on the *errno.*

3.15   SCR: Scripts

- `(SCR, 8) Invalid script termination for controller` *<controller>* .

   The controller script is not correct or invalid.

   Action:
   > Check the controller script.

- `(SCR, 9) REASON: failed to execute script` *<script>* `with` `resource` *<resource>* : *errorreason* .

   The detector script is not good or the format is not good.

   Action:
   > Check the detector script.

- `(SCR, 20) The attempt to shut down the cluster host` *host* `has` `failed:` *errorreason* .

   The cluster host could not be killed because of one of the following reasons:
   – Script exited with a non-zero status.
   – Script exited due to signal caught.
   – Other unknown failure.

   Action:
   > Verify the status of the node, make any necessary corrections to the script, potentially correct the node state manually if possible and issue appropriate `'hvutil -{o, u}'` as needed.

- `(SCR, 21) Failed to execute the script` *<script>* `, errno =` *<errno>* `, error reason:` *<errorreason>* .

   If the *script* cannot be executed, this message is printed along with the *errorreason*.

   Action:
   > Take action based on the *errorreason*.

- (SCR, 26) The sdtool notification script has failed with status *status* after dynamic modification.

    After dynamic modofication, the Shutdown Facility is notified via `sdtool` about the changes in the current configuration. If `sdtool` exits abnormally, then the base monitor must exit.

    Action:
    Verify that `sdtool` and the Shutdown Facility are operating properly.

3.16 SWT: Switch requests (hvswitch command)

- (SWT, 4) *object* is online locally, but is also online on *onlinenode* .

    If the object *object* is online on more than one host, this message is printed.

    Action:
    Make sure that the object *object* is online on only one host in the cluster.

- (SWT, 20) Could not remove host <*hostname*> from local priority list.

    A host has left the cluster, but RMS was unable to remove the corresponding entry from its internal Priority List. This is an internal problem in the program stack and memory management.

    Action:
    Contact field support.

- (SWT, 25) *objectname* : outstanding switch request of dead host was denied; cluster may be in an inconsistent condition!

    A host died during the processing of a switch request. If the host that takes over the responsibility for that particular userApplication tried to proceed with the partly-done switch request, but another host does not agree. This indicates a severe cluster inconsistency and critical internal error.

    Action:
    Contact field support.

● (SWT, 26) *object*: `dead host` *<hostname>* `was holding an unknown lock. Lock will be skipped!`

> This message appears when the dead host *<hostname>* was holding a lock that is unknown to the new responsible host.

Action:
> Allow time for the cluster to cleanup.

● (SWT, 45) `hvshut aborted because of a busy uap` *<userapplication>*`.`

> The `hvshut` request was aborted because the application is busy.

Action:
> Do not shut down RMS when its applications are busy. Make sure the application finishes its processing before shutting down RMS.

● (SWT, 46) `hvshut aborted because modification is in progress.`

> The `hvshut` request was aborted because dynamic modification is in progress.

Action:
> Do not shut down RMS while dynamic modification is in progress. Wait until dynamic modification finishes before shutting down RMS.

3.17   SYS: SysNode objects

● (SYS, 1) `Error on SysNode:` *object*`. It failed to send the kill success message to the cluster host:` *host*`.`

> When a cluster host is killed, the host requested the kill must send a success message to the surviving hosts. This message appears in the switchlog when this message send fails.

Action:
> Make sure the cluster and network conditions are such that the message can be sent across the network.

- `(SYS, 8) RMS failed to shut down the host` *host* `via a Shutdown Facility, no further kill functionality is available. The cluster is now hung.`

  This message appears when the RMS was sending a kill request to the Shutdown Facility and did not get the elimination acknowledgement.

  Action:

  Refer to the manuals of the ShutDown Facility to find out what was going wrong with the host elimination. Check the actual status of the remote host and invoke the appropriate 'hvutil -u' or 'hvutil -o' command to resolve the RMS hang state.

- `(SYS, 13) Since this host` *\<hostname>* `has been online for no more than` *time* `seconds, and due to the previous error, it will shut down now.`

  This message appears when the checksum of this host is different from the hosts in the cluster (one of the possible reasons).

  Action:

  Check the configuration in all the cluster hosts and verify that same configuration is running on all of them.

- `(SYS, 14) Neither automatic nor manual switchover will be possible on this host until` *\<detector>* `detector will report offline or faulted.`

  When different configurations are encountered in a cluster where one host is offline and the other is online.

  Action:

  Run the same configuration in a single cluster or different clusters do not have common hosts.

- `(SYS, 15) The uname() system call returned with Error. RMS will be unable to verify the compliance of the RMS naming convention!`

  This message appears when uname() system call returned with a non-zero value.

  Action:

  Make sure that the SysNode name is valid and restart RMS as needed.

- (SYS, 17) The RMS internal SysNode name "*sysnode*" is ambiguous with the name "*name*". Please adjust names compliant with the RMS naming convention "SysNode = `uname -n`RMS"

    The RMS naming convention '_sysnodename_ = `uname -n`RMS' is intended to allow use of the CF-name with and without trailing "RMS" whenever an RMS command expects a SysNode reference. This rule creates an ambiguity if one SysNode is named "*xxx*RMS" and another is named "*xxx*", because '_rms_command_ *xxx*' could refer to either SysNode. Therefore, ambiguous SysNode names are not be allowed.

    Action:
    Use non-ambiguous SysNode names and adhere to the RMS naming conventions.

- (SYS, 48) Remote host *<hostname>* replied the checksum *<remotechecksum>* which is different from the local checksum *<localchecksum>*. The sysnode of this host will not be brought online.

    This message appears when the remote host *<hostname>* is running different configuration than the local host or different loads of RMS package are installed on these hosts.

    Action:
    Make sure all the hosts are running the same configuration and the configuration is distributed on all hosts. Make sure that same RMS package is installed on all hosts (same load).

- (SYS, 49) Since this host *<hostname>* has been online for more than *time* seconds, and due to the previous error, it will remain online, but neither automatic nor manual switchover will be possible on this host until *<detector>* detector will report offline or faulted.

    This message appears when the checksum of this host is different from the hosts in the cluster (one of the possible reasons).

    Action:
    Check the configuration in all the cluster hosts and verify that same configuration is running on all of them.

- (SYS, 50) Since this host *<hostname>* has been online for no more than *time* seconds, and due to the previous error, it will shut down now.

  This message appears when the checksum of this host is different from the hosts in the cluster (one of the possible reasons).

  Action:

  Check the configuration in all the cluster hosts and verify that same configuration is running on all of them.

- (SYS, 84) Request <hvshut -a> timed out. RMS will now terminate! Note: some cluster hosts may still be online!

  This message appears when the default timeout for the hvshut command expired and some of the hosts are still running.

  Action:

  Adjust the default timer by setting RELIANT_SHUT_MIN_WAIT to a value that is large enough to allow a shutdown on all hosts. Check if shutdown failed due to internal problems (e.g., a failure of an OfflineScript prevented a userApplication from going Offline).

- (SYS, 90) *hostname* internal WaitList addition failure! Cannot set timer for delayed detector report action!

  System Error.

  Action:

  Contact field support.

- (SYS, 93) The cluster host *hostname* is not in the Wait state. The hvutil command request failed!

  This message appears when the user issues the hvutil command ('hvutil -o' or 'hvutil -u') and the cluster host *<nodename>* is not in the Wait state.

  Action:

  Reissue 'hvutil -{o, u}' only when the host is in a Wait state.

- (SYS, 94) The last detector report for the cluster host *hostname* is not online. The hvutil command request failed!

  This message appears when the user issues the `hvutil` command (`'hvutil -o sysnode'`) to clear the Wait state of the `SysNode` and the `SysNode` is still in Wait state because the last detector report for the cluster host <*hostname*> is not Online i.e. the `SysNode` might have transitioned to Wait state not from Online but from some other state.

  Action:
  Issue `'hvutil -o'` only when the host is in a Wait state that has transitioned from the Online state.

- (SYS, 97) Cannot access the NET_SEND_Q queue.

  When a new host comes Online, the other hosts in the cluster try to determine if the new host has been started with `-C` option. The host that has just come online uses the queue NET_SEND_Q to send the necessary information to the other hosts in the cluster. If this host is unable to access the queue NET_SEND_Q this message is printed.

  Action:
  Contact field support.

- (SYS, 98) Message send failed in SendJoinOk.

  When a new host comes Online, the other hosts in the cluster try to determine if the new host has been started with `-C` option. The host that has just come online uses the queue NET_SEND_Q to send the necessary information to the other hosts in the cluster. If this host is unable to send the necessary information to the other hosts in the cluster, this message is printed.

  Action:
  Check if there is a problem with the network.

- (SYS, 100) The value of the attribute <*attr*> specified for SysNode <*sysnode*> is <*invalidvalue*> which is invalid. Ensure that the entry for <*attr*> is resolvable to a valid address.

  The value of *attr* is not resolvable to a valid network address.

  Action:
  Ensure that a valid interface is specified for *attr*.

3.18   UAP: userApplication objects

● (UAP, 1) Request to go online will not be granted for appli-
cation <*userapplication*> since the host <*sysnode*> runs a
different RMS configuration.

  This message appears when the request is done for an application
  <*userapplication*> to go Online but the host <*sysnode*> is running a
  different configuration.

  Action:
  Make sure that the user is running the same configuration.

● (UAP, 5) *object*: cmp_Prio: *list*.

  The priority list *list* has invalid entries.

  Action:
  Contact field support.

● (UAP, 6) Could not add new entry to priority list.

  Critical internal error.

  Action:
  Contact field support.

● (UAP, 7) Could not remove entries from priority list.

  Critical internal error.

  Action:
  Contact field support.

● (UAP, 8) *object*: cpy_Prio failed, source list corrupted.

  This message appears when either the PriorityList is empty or the list is
  corrupted. Critical internal error.

  Action:
  Contact field support.

- (UAP, 9) *object*: `Update of PriorityList failed, cluster may be in inconsistent condition.`

    If a contract that is supposed to be present in the internal list does not exist, this message is printed. The cluster may be in an inconsistent condition.

    Action:
    Contact field support.

- (UAP, 15) *sysnode*: `PrepareStandAloneContract() processing unknown contract.`

    This message appears when there is only one application <*sysnode*> Online and has to process a contract that is not supported. Critical internal error.

    Action:
    Contact field support.

- (UAP, 16) *object*::`SendUAppLockContract: local host doesn't hold a lock -- Contract processing denied.`

    This message appears when the contract is processed by the local host, which does not have the lock for that application contract. Critical internal error.

    Action:
    Contact field support.

- (UAP, 19) *object*::`SendUAppLockContract: LOCK Contract cannot be sent.`

    This message appears when the LOCK contract cannot be sent over the network.

    Action:
    The network may be down.

- (UAP, 21) *object*::SendUAppUnLockContract: UNLOCK Contract cannot be sent.

    This message appears when the UNLOCK contract cannot be sent over the network.

    Action:
    The network may be down.

- (UAP, 22) *object* unlock processing failed, cluster may be in an inconsistent condition!

    This message appears when the local node receives a UNLOCK contract but is unable to perform the follow up processing that was committed in the contract.

    Action:
    Contact field support.

- (UAP, 23) *object* failed to process UNLOCK contract.

    A host was unable to propagate the received UNLOCK contract, e.g., because of networking problems or memory problems.

    Action:
    This message should appear with an additional ERROR message specifying the origin of the problem. Refer to the ERROR message.

- (UAP, 24) Deleting of local contractUAP object failed, cannot find object.

    This message appears when the local contract node has completed the contract and has sent it to the local node but the local node could not able to find it.

    Action:
    Contact field support.

- (UAP, 27) *object* received a DEACT contract in state: *state*.

    The correspondent userApplication on a remote host is in the DeAct state, but the local userApplication is not. Critical internal error.

    Action:
    Contact field support.

- (UAP, 28) *object* `failed to update the priority list. Cluster may be in an inconsistent state.`

  When the local host receives a contract for unlocking the hosts in the cluster with respect to a particular operation, if the local host finds that a particular host has died, it updates its priority list to reflect this, but if it is unable to perform this operation due to some reason, this message is printed. This indicates a critical internal problem in memory management.

  Action:
  Contact field support.

- (UAP, 29) *object*`: contract data section is corrupted.`

  This message appears when the application is unable to read the data section of the contract.

  Action:
  Contact field support.

- (UAP, 32) *object* `received unknown contract.`

  This message appears when the application unable to unlock the contract as it was unable to find the kind of contract request in its code that it expected. Critical internal error.

  Action:
  Contact field support.

- (UAP, 33) *object* `unknown task in list of outstanding contracts.`

  This message appears when a userApplication object finds a task in the list of outstanding contracts but unable to process it as it could not able to find the kind of contract request in its code. Critical internal error.

  Action:
  Contact field support.

- (UAP, 35) *object*: inconsistency occurred. Any further switch request will be denied (except forced requests). Clear inconsistency before invoking further actions!

     This message appears when the state of the application is offline or standby and some of the resources are online and faulted.

     Action:
     Clear the inconsistency by the appropriate command (usually 'hvutil – c').

- (UAP, 41) cannot open file *filename*. Last Online Host for userApplication cannot be stored into non-volatile device.

     File open error.

     Action:
     Check the reliant path.

- (UAP, 42) found incorrect entry in status file: >*entry*<

     This message appears when the status_info file has incorrect entry in it. This should occur only if the status info file was edited manually.

     Action:
     Check the status info file for manual incorrect entries. If this is not the case contact field support.

- (UAP, 43) *<object>*: could not insert *<host>* into local priority list.

     Action:
     Critical error. Contact field support.

- (UAP, 44) *<object>*: could not remove *<host>* from local priority list.

     Action:
     Critical error. Contact field support.

- (UAP, 45) *<object>*: could not remove *<host>* from priority list.

     Action:
     Critical error. Contact field support.

3.19   US: us files

● (US, 5) The cluster host *hostname* is no longer reachable!
Please check the status of the host and the ethernet
connection.

> This message is a result of one cluster host detecting that another cluster
> host *hostname* is no longer reachable. In other words, this cluster host
> sees the other host *hostname* as faulted. The other host *hostname* may
> have gone down, or there may some problem with the cluster inter-
> connect.

Action:
> See if the host *hostname* is indeed dead. If not, see if there is a problem
> with the network connection.

● (US, 6) RMS has died unexpectedly on the cluster host
hostname!

> When the detector on the local host detects that the host *hostname* has
> transitioned from Online to Offline unexpectedly, it prints this message to
> the switchlog and then it attempts to kill the host *hostname*.

Action:
> Check the syslog on the host *hostname* to find out the reason why it has
> gone down.

● (US, 31) FAULT REASON: Resource *resource* transitioned to a
Faulted state due to a detector report.

> This message is printed when a detector unexpectedly reports the
> Faulted state.

Action:
> Check to see if there is any problem with *resource*.

3.20   WLT: Wait list

● (WLT, 1) REASON: Resource *resource*'s *script* (*scriptexecd*) has
  exceeded the ScriptTimeout of *timeout* seconds.

  The detector script for the resource has exceeded the ScriptTimeout
  limit.

  Action:
  Make sure that timeout is large enough to execute the script.

● (WLT, 3) Cluster host *hostname*'s Shutdown Facility invoked
  via (*script*) has not finished in the last *time* seconds. An
  operator intervention is required!

  The Shutdown Facility that is killing host *hostname* has not terminated yet.
  Operator intervention may be required. This message will appear period-
  ically (with the period equal to the node's ScriptTimeout value), until
  either the script terminates on its own, or until the script is terminated by
  the Unix kill command. If terminated by the kill command, the host
  being killed will not be considered killed.

  Action:
  Wait until the script terminates, or terminate the script using kill
  command if the script cannot terminate on its own.

● (WLT, 5) CONTROLLER FAULT: Controller <*object*> has propagated
  <*request*> request to its controlled application(s) <*applica-*
  tions>, but the request has not been completed within the
  period of <*timeout*> seconds.

  When controller propagates its requests to the controlled applications, it
  is waiting for the completion of the request for a period of time sufficient
  for the controlled applications to process the request. When the request
  if not completed within this period, controller faults.

  Action:
  Fix the controller's scripts and/or scripts of the controlled applications, or
  repair resources of the controlled applications. For user defined
  controller scripts increase their ScriptTimeout values.

3.21 WRP: Wrappers

● (WRP, 1) Failed to set script to TS.

  The script could not be made into a time sharing process.

  Action:
  Take action based on the reason.

● (WRP, 2) Illegal flag for process wrapper creation.

  Action:
  Critical error. Contact field support.

● (WRP, 3) Failed to execv: *command*.

  This message could occur in any of the following scenarios:
  – A detector cannot be started because RMS is unable to create the detector process with the command *command*.
  – 'hvcm -a' has been invoked and the RMS base monitor cannot be started on the individual hosts comprising the cluster with the command *command*.
  – A script cannot be started because RMS is unable to create the script process with the command *command*.
  RMS shuts down on the node where this message appears and returns an error number *errno*, which is the error number returned by the operating system.

  Action:
  Consult the system manual pages or the appendix of this manual for the explanation for error number *errno* and see if the cause is evident. If not, contact field support.

● (WRP, 4) Failed to create a process: *command*.

  This message could occur in any of the following scenarios:
  – A detector cannot be started because RMS is unable to create the detector process to execute the command *command*.
  – 'hvcm -a' has been invoked and the RMS base monitor cannot be started on the individual hosts comprising the cluster with the command *command*.
  – A script cannot be started because RMS is unable to create the script process with the command *command*.

RMS shuts down on the node where this message appears and returns an error number *errno*, which is the error number returned by the operating system.

Action:
Consult the system manual pages or the appendix of this manual for the explanation for error number *errno* and see if the cause is evident. If not, contact field support.

● (WRP, 5) No handler for this signal event *<signal>*.

There is no signal handler associated with the signal *signal*.

Action:
Contact field support.

● (WRP, 6) Cannot find process (pid=*processid*) in the process wrappers.

Action:
Critical error. Contact field support.

● (WRP, 7) getservbyname failed for service name: *servicename*.

Action:
Critical error. Contact field support.

● (WRP, 8) gethostbyname failed for remote host: *host*.

Action:
Critical error. Contact field support.

● (WRP, 9) Socket open failed.

This message occurs if RMS is unable to create a datagram endpoint for communication.

Action:
Contact your System Administrator.

● (WRP, 10) connect to server failed.

Action:
Critical error. Contact field support.

● (WRP, 11) `Message send failed, queue id` *<queueid>*`, process` *<process>*`,` *<name>*`, to host` *<host>*`.`

RMS exchanges messages between processes and hosts to maintain inter-host communication. If the delivery of a message has failed then this error is printed. This can occur if one or more hosts in the cluster are not active or if there is a problem with the network.

Action:
(i) Check the other hosts in the cluster. If any are not alive, check the switchlog for information regarding why RMS has died on those hosts. Perform the following steps in order:
1. `'hvdisp -a'`
2. In the output of step (a) check if the state of any of the resources whose type is `SysNode` is offline. If so, that means that RMS is not running on that node.
3. Check the switchlogs of all the nodes that are offline to determine the reason why RMS on that node is not active.
(ii) If the other hosts that are part of the cluster are alive then that means there is some problem with the network.

● (WRP, 12) `Failed to bind port to socket.`

This could occur if RMS is unable to bind the endpoint for communication.

Action:
Contact field support.

● (WRP, 14) `No available slot to create a new host instance.`

When the base monitor for RMS starts up, it creates a slot in an internal data structure for every host in the cluster.
When `hvdet_node` is started up, RMS sends it a list of the `SysNode` objects that are put into different slots in the internal data structure. If the data structure has run out of slots (16) to put the `SysNode` name in, this message is printed.

Action:
Contact field support.

- (WRP, 15) gethostbyname(*hostname*): host name should be in /etc/hosts

    When the hostname *hostname* specified as a `SysNode` does not have an entry in `/etc/hosts`, this message is printed to the switchlog.

    Action:
    Correct the host name *hostname* to be an entry in `/etc/hosts`.

- (WRP, 16) No available slot for host *hostname*

    When RMS has run out of slots for the cluster interfaces (64), this message is printed along with the host name *hostname* for which this happened.

    Action:
    Contact field support.

- (WRP, 17) Size of integer or IP address is not 4-bytes

    Critical internal error.

    Action:
    Contact field support.

- (WRP, 18) Not enough memory in *<processinfo>*

    Action:
    Critical error. Contact field support.

- (WRP, 23) The child process *<cmd>* with pid *<pid>* could not be killed due to errno *<errno>*, reason: *reason*.

    The child process with pid *pid* could not be killed due to reason: *reason*.

    Action:
    Take action based on the reason *reason*.

- (WRP, 24) Unknown flag option set for 'killChild'.

    The killChild routine accepts one of the 2 flags: KILL_CHILD and DONTKILL_CHILD. If an option other than these two has been specified, this message is printed.

    Action:
    Please contact professional services.

● (WRP, 25) Child process *<cmd>* with pid *<pid>* has exceeded its timeout period. Will attempt to kill the child process.

    The child process *cmd* has exceeded its timeout period.

Action:
    Please contact professional services.

● (WRP, 28) RMS monitor has encountered an irregular sequence of timer interrupts, off by *<offset>* seconds. This may have been caused by a manual OS time change, or by an unusually high OS performance load, or by some other OS condition. If this error appears frequently, then normal RMS operations can no longer be guaranteed; it can also lead to a loss of heartbeats with remote hosts and to an elimination of the current host from the cluster.

    The RMS base monitor keeps track of the regularity of its timer interrupts that are supposed to occur every second. If the interrupts become irregular due to a high load, manual time change, or any other reason, the above notice is printed. If the discrepancy value becomes too high, or if this error appears frequently, then this might lead to a malfunction of the RMS base monitor, which can cause a loss of High Availability.

Action:
    Do not attempt to change the system date/time by any significant value while RMS is running. Raise the priority of the RMS base monitor to ensure that it has enough CPU time to perform its operations during a high load.

● (WRP, 29) RMS on the local host has received a message from host *host*, but the local host is unable to resolve the sending host`s address. This could be due to a misconfiguration. This message will be dropped. Further such messages will appear in the switchlog.

    RMS on the local host has received a message from host *host* whose address is not resolvable by the local host.

Action:
    Make sure that the local host is able to resolve the remote host *host*'s address by checking for any misconfigurations.

- (WRP, 30) RMS on the local host has received a message from host *host*, but the local host is unable to resolve the sending host's address. This message will be dropped. Please check for any misconfiguration.

  RMS on the local host has received a message from host *host* whose address is not resolvable by the local host.

  Action:

  Make sure that the local host is able to resolve the remote host *host*'s address by checking for any misconfigurations.

- (WRP, 31) RMS has received a message from host *host* with IP address *receivedip*. The local host has calculated the IP address of that host to be *calcip*. This may be due to a misconfiguration in /etc/hosts. Further such messages will appear in the switchlog.

  The local host has received a message from host *host* with IP address *receivedip*, which is different from the locally calculated IP address for that host.

  Action:
  Check /etc/hosts for any misconfiguration.

- (WRP, 32) RMS has received a message from host *host* with IP address *receivedip*. The local host has calculated the IP address of that host to be *calcip*. This may be due to a misconfiguration in /etc/hosts.

  The local host has received a message from host *host* with IP address *receivedip*, which is different from the locally calculated IP address for that host. This message will be printed in the switchlog for every 25 such messages that have been received as long as the number of received messages is less than 500, if not this message is printed for every 250th such message received.

  Action:
  Check /etc/hosts for any misconfiguration.

- (WRP, 33) Error while creating a message queue with the key *<id>*, errno = *<errno>*, explanation: *<explanation>*.

  An abnormal OS condition occurred while creating a message queue.

  Action:
  Check OS conditions that affect memory allocation for message queues, such as the size of swap space, the values of parameters `msgmax`, `msgmnb`, `msgmni`, `msgtql`. Check if the maximum number of message queues have already been allocated.

- (WRP, 34) Cluster host *host* is no longer in time sync with local node. Sane operation of RMS can no longer be guaranteed. Further out-of-sync messages will appear in the syslog.

  The time on *host* is not in sync with the time on the local node.

  Action:
  Sync the time on *host* with the time on the local node.

- (WRP, 35) Cluster host *host* is no longer in time sync with local node. Sane operation of RMS can no longer be guaranteed.

  The time on the cluster host *host* differs significantly (> 25 seconds) from the local node.

  Action:
  Make sure that all the cluster hosts are in time sync.

- (WRP, 52) The operation *func* failed with error code *errorcode*.

  The operation *func* failed with error code *errorcode*.

  Action:
  Contact field support.

4 Fatal error messages

This chapter contains a detailed list of all fatal RMS error messages that appear in the switchlog. Most messages are accompanied by a description of the probable cause(s) and a suggested action to correct the problem. In some cases, the description or action is self-evident and no further information is necessary.

Some messages in the listings that follow contain words printed in *italics*. These words are placeholders for values, names, or strings that will be inserted in the actual message when the error occurs.

RMS error code description

A prefix in each message contains an error code and message number identifying the RMS component that detected the problem. You may need to provide this prefix to support engineers who are diagnosing your problem. The following list summarizes the possible error codes and the associated component:

    ADC: Admin configuration
    ADM: Admin, command, and detector queues
    BM: Base monitor
    CML: Command line
    CMM: Communication
    CRT: Contracts and contract jobs
    DET: Detectors
    INI: init script
    MAS: Master process
    MIS: Miscellaneous
    QUE: Message queues
    SCR: Scripts
    SYS: SysNode objects
    UAP: userApplication objects
    US: us files
    WLT: Wait list
    WRP: Wrappers

4.1 ADC: Admin configuration

● (ADC, 16) Because some of the global environmental variables were not set up in hvenv file, RMS cannot startup. Shutting down.

> All of the global environment variables RELIANT_LOG_LIFE, RELIANT_SHUT_MIN_WAIT, HV_CHECKSUM_INTERVAL, HV_LOG_ACTION_THRESHOLD, HV_LOG_WARNING_THRESHOLD, HV_WAIT_CONFIG and HV_RCSTART have to be set in hvenv in order for RMS to function properly. If some of them have not been set, RMS exits with exit code 1.

Action:
> Set the values of all the environment variables in hvenv.

● (ADC, 21) Because some of the local environmental variables were not set up in hvenv file, RMS cannot startup. Shutting down.

> If some of the local environment variables have not been set in the hvenv file, RMS prints this message and exits with exit code 1.

Action:
> Make sure that all the local environment variables have been set to an appropriate value in the hvenv file.

● (ADC, 69) RMS will not start up – previous errors opening file.

> The previous error was a failure to open the file needed for dynamic startup. The base monitor will exit.

Action:
> Verify the file existence and reissue dynamic startup request.

4.2   ADM: Admin, command, and detector queues

● (ADM, 1) cannot open admin queue.

> RMS uses UNIX message queues for interprocess communication. The admin queue is one such queue used for communication between utilities like hvutil, hvswitch, etc. If there is a problem opening this queue, then this message is printed and RMS exits with exit code 3.

Action:
> Contact field support.

● (ADM, 2) RMS will not start up — errors in configuration file.

> When RMS is starting up, it performs dynamic modification under the hood, if during this phase it encounters errors in its configuration file, RMS exits with exit code 23.

Action:
> Make sure there are no errors in the configuration file based on the error messages printed prior to the above message in the switchlog.

4.3   BM: Base monitor

● (BM, 3) Usage: *progname* [-c config_file] [-m] [-h time] [-l level] [-n]

> An attempt has been made to start RMS in a way that does not conform to its expected usage. This message is printed to the switchlog indicating the arguments, and RMS exits with exit code 3.

Action:
> Start RMS with the right arguments.

● (BM, 49) Failure calculating configuration checksum.

> During dynamic reconfiguration, RMS calculates the configuration checksum by using /usr/bin/sum. If this fails, then this message is printed and RMS exits with the exit code 52.

Action:

Check if `/usr/bin/sum` is available.

● (BM, 51) The RMS-CF interface is inconsistent and will require operator intervention. The routine "*routine*" failed with errno *errno* - "*errorreason*"

While setting up CF, if RMS encounters a problem in the routine *routine* that can either be "dlopen" or "dlsym", it exits with exit code 95 or 94 respectively. The *errorreason* gives the reason for the error.

Action:

Contact field support.

● (BM, 58) Not enough memory -- RMS cannot continue its operations and is shutting down.

This is a generic message that is printed to the switchlog before RMS discontinues its functioning because it does not have enough memory for it to operate.

Action:

Contact field support.

● (BM, 67) An error occurred while writing out the RMS configuration after dynamic modification. RMS is shutting down.

Upon concluding dynamic modification, RMS dumps out its current configuration into a file `/var/tmp/config.us`. If this cannot be done, RMS cannot recalculate configuration's checksum. Therefore, it shuts down.

Action:

The previous message in switchlog explains why RMS has not been able to write down the configuration file. Correct the host environment according to the description, or contact field support.

● (BM, 69) Some of the OS message queue parameters msgmax=
  <msgmax>, msgmnb=*<msgmnb>*, msgmni=*<msgmni>*, msgtql=*<msgtql>*
  are below lower bounds *<hvmsgmax>*, *<hvmsgmnb>*, *<hvmsgmni>*,
  <hvmsgtql>. RMS is shutting down.

  One or more of the system defined message queue parameters is not
  sufficient for correct operation of RMS. RMS shuts down with exit code
  28.

  Action:
  Change the OS message queue parameters and reboot the OS before
  restarting RMS.

● (BM, 82) A message to host *<remotehost>* failed to reach that
  host after *<count>* delivery attempts. Communication with that
  host has been broken. Therefore, RMS monitor on this host
  <localhost> is going down.

  A communication breakdown prevented delivery of a message between
  the local and remote RMS monitors. In this case the local monitor exits.

  Action:
  Make sure *remotehost* is up and that communication between the two
  hosts is possible. Use standard tools such as ping and make sure that
  the local root account can rlogin or rsh to the remote host. After
  communication has been re-established, restart the local RMS monitor.

● (BM, 89) The SysNode length is *length*. This is greater than
  the maximum allowable length of *maxlength*. RMS will now shut
  down.

  The SysNode name length is greater than the maximum allowable
  length.

  Action:
  Ensure that the length of the SysNode name is less than *maxlength*.

4.4 CML: Command line

● (CML, 14) ###ERROR Unable to find or Invalid configuration file.### #####CONFIGURATION MONITOR exits !!!!!######

> The configuration file specified for RMS is non-existent. RMS exits with exit code 1.

Action:
> Specify a valid configuration file for RMS to function.

4.5 CMM: Communication

● (CMM, 1) Error establishing outbound network communication.

> If there is an error in creating outbound network communication, this message is printed, and RMS exits with exit code 12.

Action:
> System error. Contact field support.

● (CMM, 2) Error establishing inbound network communication.

> If there is an error in creating inbound network communication, this message is printed, and RMS exits with exit code 12.

Action:
> System error. Contact field support.

4.6 CRT: Contracts and contract jobs

● (CRT, 6) Fatal system error in RMS. RMS will shut down now. Please check the bmlog for SysNode information.

> A system error has occurred within RMS.

Action:
> Contact field support.

4.7   DET: Detectors

● (DET, 8) Failed to create DET_REP_Q.

> If RMS is unable to create the Unix Message queue DET_REP_Q for
> communication between a detector and itself, this message is printed,
> and RMS exits with exit code 12.

Action:
> Contact field support.

● (DET, 9) Message send failed in detector request Q: *queue*.

> During hvlogclean, the detector request queue *queue* is used for
> sending information to the detector from the base monitor. If there is a
> problem in communication, this message is printed, and RMS exits with
> exit code 12.

Action:
> Contact field support.

● (DET, 16) Cannot create gdet queue of kind g*kind*.

> Each of the generic detectors has a message queue that it uses to
> communicate with the base monitor. If there is a problem creating a
> queue for a detector of kind *kind*, this message is printed, and RMS exits
> with exit code 12.

Action:
> Contact field support.

● (DET, 18) Error reading hvgdstartup file. Error message:
errorreason.

> When the RMS base monitor tries starting up the generic detectors, it
> parses the hvgdstartup file for detector information. If RMS encounters
> an error while reading this file, it prints this message along with the
> reason *errorreason* for the failure. RMS then exits with exit code 26.

Action:
> Contact field support.

4.8 INI: init script

- (INI, 4) InitScript does not have execute permission.

    InitScript exists, but cannot be executed.

    Action:
    make InitScript executable.

- (INI, 7) *sysnode* must be in your configuration file.

    If the local SysNode *sysnode* is not part of the configuration file, this message is printed, and RMS exits with exit code 23.

    Action:
    Make sure that the local SysNode *sysnode* is part of the configuration file.

- (INI, 10) InitScript has not completed within the allocated time period of *timeout* seconds.

    InitScript was still running when the timeout limit allocated for its execution has expired. The timeout limit is the lesser of the values defined in the environment variable SCRIPTS_TIME_OUT or 300.

    Action:
    Increase the timeout value, or correct the conditions that lead to timeout during script execution.

- (INI, 11) InitScript failed to startup, errno *errno*, reason: *reason*.

    An error occurred during startup of InitScript. The errno code *<errno>* and reason *<reason>* are presented in the message.

    Action:
    Correct the erroneous host condition for InitScript to be able to start up.

- (INI, 12) InitScript returned non-zero exit code *exitcode*.

    InitScript completed with a non-zero exit code *<exitcode>*.

    Action:
    Correct the erroneous host condition for InitScript to be able to return a zero exit code, or fix the InitScript itself.

● `(INI, 13) InitScript has been stopped.`

    `InitScript` has been stopped.

Action:
    Correct the erroneous host condition for InitScript to run without
    stopping, or fix the `InitScript` itself.

● `(INI, 14) InitScript has been abnormally terminated.`

    `InitScript` has been abnormally terminated.

Action:
    Correct the erroneous host condition for `InitScript` to run without
    stopping, or fix the `InitScript` itself.

4.9   MAS: Master process

● `(MAS, 1) Usage: master -t interval`

    Critical internal error.

Action:
    Contact field support.

● `(MAS, 2) The RMS-CF interface is inconsistent and will
    require operator intervention. The routine "`*routine*`" failed
    with errno` *errno* `- "`*errorreason*`"`

    While setting up CF, if the master process encounters a problem in the
    routine *routine* that can either be "dlopen" or "dlsym", it exits with exit
    code 94 or 94 respectively. The *errorreason* gives the reason for the error.

Action:
    Contact field support.

● `(MAS, 6)` *detector*`: Failed to open rep_queue.`

    The process *detector* was unable to open the rep_queue.

Action:
    Contact field support.

● (MAS, 7) *detector*: `Unable to send message` *message* `to the base monitor.`

> The process *detector* was unable to send the message *message* to the base monitor.

Action:
> Contact field support.

4.10   MIS: Miscellaneous

● (MIS, 4) `The locks directory` *directory* `cannot be cleaned of all old locks files: at` *call*, `errno =` *errnonumber*, `error --` *errortext*.

> The various RMS commands like `hvdisp`, `hvswitch`, `hvutil` and `hvdump` utilize the lock files from the directory *directory* for signal handling purposes. These files are deleted after these commands are completed. The locks directory is also cleaned when RMS starts up. If they are not cleaned for some reason, this message is printed, and RMS exits with exit code 99. The *call* indicates at which stage the cleanup has failed, *errornumber* is the OS errno value, *errortext* is the OS supplied explanation for the errno.

Action:
> Make sure that the locks directory *directory* exists.

4.11   QUE: Message queues

● (QUE, 1) `Error status in ADMIN_Q.`

> Different utilities use the ADMIN_Q to communicate with the base monitor. If there is an error with this queue, this message is printed and RMS exits with exit code 3.

Action:
> Contact field support.

● (QUE, 2) Read message failed in ADMIN_Q.

The RMS base monitor was unable to extract a message of the ADMIN_Q that is used for communication between the utilities and RMS. This message is printed, and RMS then exits with exit code 3. Critical error.

Action:
Contact field support.

● (QUE, 5) Network message read failed.

If there is a problem reading a message over the network, this message is printed, and RMS exits with exit code 3.

Action:
Critical error. Contact field support.

● (QUE, 6) Network problem occurred.

A network problem occurred when transferring messages.

Action:
System error. Contact field support.

● (QUE, 11) Read message failed in DET_REP_Q.

All the detectors use the queue DET_REP_Q to communicate with the RMS base monitor. If there is a problem in reading the message of the queue, RMS prints this message and exits with exit code 15.

Action:
Contact field support.

● (QUE, 12) Error status in DET_REP_Q: *status*.

The RMS base monitor encountered a problem with the queue DET_REP_Q that is used by the different detectors to report their state. This message is printed, and RMS then exits with exit code 15.

Action:
Contact field support.

4.12   SCR: Scripts

- (SCR, 4) Failed to create a detector request queue for
  detector *detectorname*.

  If a detector request queue could not be created for detector
  detector_name, this message is printed, and RMS exits with exit code 12.

  Action:
  System problem. Contact field support.

- (SCR, 5) REQUIRED PROCESS RESTART FAILED: Unable to restart
  detector. Shutting down RMS.

  If the detector *detector* could not be restarted, this message is printed,
  and RMS exits with exit code 14. The restart could have failed for any of
  the following reasons:
  – If the detector needs to be restarted more than 3 times in one minute.
  – If there is a problem with memory allocation within RMS.

  Action:
  Contact field support.

- (SCR, 10) InitScript did not run ok. RMS is being shut down.

  RMS runs the InitScript initially. The value of InitScript is the
  value of the environment variable RELIANT_INITSCRIPT. If
  InitScript fails (e.g., exits with a non-zero code, gets a signal), then
  this message is printed and RMS shuts down with exit code 56.

  Action:
  Contact field support.

- (SCR, 12) incorrect initialization of RealDetReport;
  Shutting down RMS.

  Since the scripts are executed based on the reports of the detectors, if
  the detector reports a state other than Online, Offline, Faulted, Standby
  or NoReport, this message is printed, and RMS exits with exit code 8.

  Action:
  Make sure that the detector only reports states Online, Offline, Faulted,
  Standby or NoReport.

● (SCR, 13) ExecScript: Failed to exec script *<script>* for object *<nodename>*: errno *errno*.

> RMS has been unable to execute a script *<script>* for the object *<objectname>*. The error number *errno* returned by the operating system provides a diagnosis of the failure. RMS exits with exit code 8.

Action:
> Consult the system manual pages or the appendix of this manual for the explanation for error number *errno* and see if the cause is evident. If not, contact field support.

● (SCR, 28) Manual Mode request failed to add satellite node *node* to AS DEACTSAT list.

> Critical error.

Action:
> Contact field support.

● (SCR, 29) Manual Mode request failed to remove satellite node *node* from AS DEACTSAT list.

> Critical error.

Action:
> Contact field support.

4.13   SYS: SysNode objects

● (SYS, 33) The RMS cluster host *<hostname>* does not have a valid entry in the /etc/hosts file. The lookup function gethostbyname failed. Please change the name of the host to a valid /etc/hosts entry and then restart RMS.

> If the lookup function gethostbyname searches the file /etc/hosts to get information about the host *hostname*, but is unable to find a valid entry for it, this message is printed and RMS exits with exit code 114.

Action:
> Make sure that the host name *hostname* has a valid entry in /etc/hosts and restart RMS.

- (SYS, 52) SysNode *sysnode*: error creating necessary message
  queue NODE_REQ_Q...exiting.

  When RMS encounters a problem in creating the NODE_REQ_Q, this
  message is printed, and RMS exits with exit code 12.

  Action:
  Contact field support.

4.14   UAP: userApplication objects

- (UAP, 36) *object*: double fault occurred, but Halt attribute
  is set. RMS will exit immediately in order to allow a
  failover!

  When the Halt attribute is set for an object and a double fault occurs, then
  RMS will exit with code 96 on that node.

  Action:
  Contact field support.

4.15   US: us files

- (US, 1) RMS will not start up – fatal errors in configuration
  file.

  Errors were found in the configuration file that prevented RMS startup.
  This is usually caused by manual editing or distribution of the configu-
  ration file.

  Action:
  Use only PCS or the Wizard Tools to create and activate your configu-
  ration. If you have used only the standard tools and this error persists,
  contact field support.

- (US, 42) A State transition error occured. See the next
  message for details.

  A state transition error occured in the course of RMS state transitons.
  Details of the error are printed in the subsequent lines.

Action:

> Save the error description and contact field support.

4.16   WLT: Wait list

● `(WLT, 9) sdtool notification timed out after` *`<timeout>`*
  `seconds.`

> After dynamic modofication, the Shutdown Facility is notified via sdtool
> about the changes in the current configuration. If this notification does
> not finish within the period specified by the local `SysNode Script-`
> `Timeout` value, the base monitor must exit.

Action:

> Verify that sdtool and Shutdown Facility are properly operating. Increase
> the `ScriptTimeout` value if needed.

4.17   WRP: Wrappers

● `(WRP, 40) The length of the` *`type`* `name specified for the host`
  `host` `is` *`<length>`* `which is greater than the maximum allowable`
  `length` *`<maxlength>`*`. RMS will exit now.`

> The length of the interconnect name is greater than the maximum value.

Action:

> Make sure that the interconnect name is less than the maximum value of
> *maxlength.*

● `(WRP, 44) Not enough slots left in the wrapper data`
  `structure to create new entries.`

> If RMS is started up with a configuration containing more than the
> supported number of sysnodes, this could happen and bm will exit.

Action:

> Make sure that RMS is not started on more than the supported number
> of sysnodes.

- (WRP, 45) The SysNode to the CIP name mapping for *<sysnode>* has failed.

    The RMS-CF-CIP mapping for *sysnode* has failed.

    Action:
    Make sure that the /etc/cip.cf file has valid entries.

- (WRP, 46) The RMS-CF interface is inconsistent and will require operator intervention. The routine "*routine*" failed with error code *errorcode* - "*errorreason*".

    This is a generic message indicating that the execution of the routine *routine* failed due to the reason *errorreason* and hence the RMS-CF interface is inconsistent.

    Action:
    Contact field support.

- (WRP, 47) The RMS-CF-CIP mapping cannot be determined for any host as the CIP configuration file *<configfilename>* cannot be opened. Please verify that all the entries in *<config-filename>* are correct and that CF and CIP are fully configured.

    The CIP configuration file *configfilename* could not be opened for reading.

    Action:
    Please ensure that the configuration file *configfilename* exists.

- (WRP, 48) The RMS-CF-CIP mapping cannot be determined for any host as the CIP configuration file *<configfilename>* has missing entries. Please verify that all the entries in *<configfilename>* are correct and that CF and CIP are fully configured.

    CIP configuration file has missing entries.

    Action:
    Make sure that the CIP configuration has entries for all the RMS hosts that are running in a cluster.

5 Console error messages

This chapter contains a detailed list of all RMS error messages that appear on the console. The messages are listed here in alphabetical order; messages that begin with replaceable strings are listed first. Most messages are accompanied by a description of the probable cause(s) and a suggested action to correct the problem. In some cases, the description or action is self-evident and no further information is necessary.

Some messages in the listings that follow contain words printed in *italics*. These words are placeholders for values, names, or strings that will be inserted in the actual message when the error occurs.

5.1 Console messages in alphabetical order

- *command1* `cannot get list of resources via` *<command2>* `from hvcm.`

    The wizards rely on `hvmod` for dynamic modification. If there is a problem executing command *command2*, this message is printed and `hvmod` exits with exit code 15.

    Action:
    Contact field support.

- *command* `failed due to errors in` *<argument>*.

    When `hvmod` has been invoked, it uses `hvbuild` internally. If there is a problem with the execution of `hvbuild`, this message is printed and `hvmod` is aborted. `hvmod` then exits with exit code 1.

    Action:
    Contact field support.

- *command*: `bad state:` *state*.

    If `hvassert` is performed for a state *state* that is not among the states that can be asserted, this message is printed and `hvassert` exits with exit code 1.

    Action:
    Make sure that the state specified for `hvassert` is assertable.

● *command*: `bad timeout:` *timeout*.

> If the timeout specified for the `hvassert` command is not a number, this message is printed and the utility exits with exit code 1.

Action:
> Specify a number for the timeout value of `hvassert`.

● *command*: `cannot open file` *filename*.

> `hvsend` is used to send messages to an object in a resource graph. It can get the list of messages to send from a file. If this file cannot be opened, this message is printed and the `hvsend` utility exits with exit code 8.

Action:
> Make sure that the file *filename* exists.

● *command*: `could not create a pipe`

> If the utility *command* could not open the tty to be written to, this message is printed and the utility exits with exit code 7.

Action:
> Contact field support.

● *command*: `failed due to undefined variable: local_host`.

> If the `hvsend` utility is unable to find the value of the environment variable `RELIANT_HOSTNAME`, this message is printed and `hvsend` exits with exit code 7.

Action:
> Make sure that `RELIANT_HOSTNAME` is defined.

● *command*: `file already exists`

> When '`hvdisp -o`' has been invoked by the user and the output file that has been specified as an argument already exists, this message is printed and `hvdisp` exits with exit code 6.

Action:
> Specify a filename that does not already exist as the argument to '`hvdisp -o`'.

- *command*: `message queue is not ready yet!`

  The command *command* relies on a message queue to transmit messages to the RMS base monitor. If this message queue is not available for some reason, this message is printed and the utility exits with exit code 3.

  Action:
  Contact field support.

- *command*: `Must be super-user to issue this command`

  This message indicates that in order to run the command *command*, the user should have root privileges.

  Action:
  Make sure that the user has root privileges before issuing the command.

- *command*: `RMS is not running`

  When the command *command* has been invoked, it checks to make sure that RMS is running, if not this message is printed and the utility exits with exit code 2.

  Action:
  Make sure that RMS is running before invoking the different utilities.

- *directory*: `cannot put message in queue`

  The various RMS commands like `hvdisp`, `hvswitch`, `hvutil` and `hvdump` utilize the lock files from the directory *directory* for signal handling purposes. These files are deleted after these commands are completed. The locks directory is also cleaned when RMS starts up. If they are not cleaned for some reason, this message is printed. RMS exits with exit code 99.

  Action:
  Make sure that the locks directory *directory* exists.

● *resource* is not in state *state*.

> If the hvassert on an object *resource* for a state *state* discovers that the resource is not in that state, this message is printed and hvassert exits with exit code 1.

Action:
> None required.

● *timestamp*: NOTICE: User has been warned of 'hvshut -f' and has elected to proceed.

> When the user invokes 'hvshut -f', and then has elected to proceed with the command then, this message is printed to confirm that 'hvshut -f' is being invoked.

Action:
> None required.

● *<command>* failed with exit code *exitcode*

> When the hvlogclean utility is invoked without the -d option, it executes the command *command*, if this command could not be executed for some reason, it returns the exit code *exitcode* and then the utility exits with exit code 6.

Action:
> Take action based on the exit code *exitcode*.

● BEWARE: 'hvshut -f' may break the consistency of the cluster.
No further action may be executed by RMS until the cluster consistency is re-established. This re-estab-lishment
includes restart of RMS on the shut down host.
Do you wish to proceed? (yes = shut down RMS / no = leave RMS running).

> This is a message asking for confirmation to proceed with 'hvshut -f'. If the user elects to proceed, yes would be the appropriate answer, and a no if there is no intention of going ahead with it.

Action:
> Respond to the prompt.

● `BEWARE: the hvreset command will result in a reinitial-`
  `ization of the graph of the specified userApplication. This`
  `affects basically the RMS state engine only. The re-initial-`
  `ization does not mean, that activities invoked by RMS so far`
  `will be made undone. Manual cleanup of halfway configured`
  `resources may be necessary.`
  `Do you wish to proceed? (yes = reset application graph / no`
  `= abort hvreset).`

  Action:
  > Respond to the prompt.

● `Can't open modification file.`

  > When `hvmod` is invoked with the `-c` option, it uses a temporary file. If this file cannot be opened for writing, this message is printed and `hvmod` exits with exit code 1.

  Action:
  > Contact field support.

● `Cannot start RMS! BM is currently running.`

  > RMS is already running on the local host.

  Action:
  > Shut down the currently running version of RMS and restart.

● `Change dest_object to` *node*`.`

  Action:
  > (None specified)

● `Command aborted.`

  > This message is printed after the user elects not to proceed with a command.

  Action:
  > None required.

● `Command timed out!`

  Action:
  > (none specified)

● Could not open localfile or could not create temporary file *filename*

> If during hvrcp, the localfile cannot be opened for reading or the temporary file *filename* cannot be opened for writing, this message is printed and hvrcp exits with exit code 7.

Action:
> Check the permissions on the localfile to make sure that it is readable.

● Could not restart RMS. RELIANT_PATH not set.

> When the detector restarts RMS, it checks the value of the environment variable RELIANT_PATH, if it cannot get the value of this variable, this message is printed.

Action:
> Make sure that RELIANT_PATH is set to an appropriate value.

● Delay *delay* seconds.....

> This is an informational message specifying the delay *delay* in seconds that hvsend has been provided.

Action:
> None required.

● DISCLAIMER: The hvdump utility will collect the scripts, configuration files, log files and any core dumps. These will be shipped out to RMS support. If there are any propri-etary files you do not want included, please exit now. Do you want to proceed? (yes = continue / no = quit)

> This message is printed on executing 'hvdump -E' and will collect the necessary information only if the answer to the above question is "yes".

Action:
> Respond to the prompt.

● DISCLAIMER: The hvdump utility will now collect the necessary information. These will be shipped to RMS support.

This message just indicates that the hvdump utility will now start collecting the information.

Action:
None required.

● Error becoming a real time process: *errorreason*

The RMS base monitor runs as a real time process (thereby giving it higher priority over other processes) on Solaris. If there is a problem in the base monitor becoming a real time process due to *errorreason*, then this message is printed.

Action:
Take action based on the reason.

● Error setting up real time parameters: *errorreason*

If there is a problem while setting up the parameters for the RMS base monitor to run as a real-time process, this message is printed along with the reason *errorreason* for the problem.

Action:
Take action based on the reason.

● Error while starting up bm on the remote host *<targethost>*: *errorreason*

When hvcm is invoked with the −s option to start RMS on a remote host *<targethost>*, if there is a problem in starting up RMS on the remote host, this message is printed along with the reason for the problem *<error-reason>*.

Action:
Take action based on the reason for the problem and reissue 'hvcm −s'.

● Error while starting up local bm: *errorreason*

Error while starting up local bm: *errorreason*

Action:
Take action based on the reason.

● `Failed to dup a file descriptor.`

> If RMS is unable to dup a file descriptor while setting the environment,
> this message appears.

Action:
> Contact field support.

● `Failed to exec the hvenv file <`*hvenvfile*`>.`

> RMS was unable to `exec` the `hvenv` environment variable file *hvenvfile*.

Action:
> Contact field support.

● `Failed to open pipe.`

> If RMS is unable to open a pipe for communication, this message is
> printed and RMS exits with exit code 1.

Action:
> Contact field support.

● `FATAL ERROR Could not restart RMS.  Restart count exceeded.`

> When the detector tries to restart RMS, it keeps track of how many times
> RMS had to be restarted. If this count has exceeded 3, then this
> message is printed.

Action:
> Contact field support.

● `FATAL ERROR: Could not restart RMS.  Restart script (`*script*`)
does not exist.`

> When the detector is unable to restart RMS because the script *script* is
> non-existent, this message is printed.

Action:
> Make sure that the script *script* exists.

● FATAL ERROR: Could not restart RMS. Failed to recreate RMS restart count file.

> When the detector tries to restart RMS, it keeps track of how many times RMS had to be restarted by writing out the necessary information out to a count file. If this file cannot be opened for writing the above message is printed.

> *Action:*
> Contact field support.

● FATAL ERROR: RMS has failed to start!

> Critical internal error.

> *Action:*
> Contact field support.

● File open failed (*path*): *errorreason*.

> If the file *path* that is used by the hvassert utility to communicate with the RMS base monitor could not be opened, this message is printed, along with the reason *errorreason* for this failure. hvassert then exits with exit code 5.

> *Action:*
> Contact field support.

● Forced shut down on the local cluster host!

> When the detector restarts the base monitor, it prints this message before proceeding.

> *Action:*
> None required.

● Fork failed.

> If RMS is unable to fork a process, it prints this message and exits with exit code 1.

> *Action:*
> Contact field support.

● hvsend: dest_object is not specified.

> If hvsend has been provided an unknown option in the input file, this
> message is printed and hvsend exits with exit code 9.

Action:
> Make sure that you specify a valid option.

● hvutil: Could not determine if RMS is running on *<targethost>*,
errno *exitcode*

> Printed when 'hvutil -A *targethost*' is called indicating that the
> command failed to ascertain whether or not RMS is running on *targethost*.
> The *exitcode* indicates a value in /usr/include/sys/errno.h.

Action:
> Depends on the *exitcode* value

● hvutil: Could not determine IP address of *<targethost>*

> The name of the cluster host could not be resolved to an IP address.

Action:
> Add an entry for *targethost* into the /etc/hosts file of all cluster hosts.

● hvutil: debug option must be a positive number for on, 0 for
off.

> When 'hvutil -L' has been invoked with a loglevel that is not one of 0
> or 1, this message is printed and it exits with exit code 6.

Action:
> Specify a valid logging level of 0 or 1 for the utility.

● hvutil: Detector time period must be greater than
minimumtime.

> If the detector time period specified as an argument with 'hvutil -t' is
> less than *minimumtime*, hvutil is aborted and exits with exit code 5.

Action:
> Invoke hvutil with a time period that is greater than *minimumtime*.

- hvutil: Failed to allocate socket

    Failed to allocate a socket to communicate with a remote host.

    Action:
    Contact professional services to determine the cause.

- hvutil: Missing /etc/services entry for "rmshb"

    An entry is missing in the /etc/services file for the RMS heartbeat.

    Action:
    Add an entry on all cluster hosts for rmshb using tcp

- hvutil: Notify string is longer than *mesglen* bytes

    Notify string is too long.

    Action:
    Notify string should not be longer than *mesglen* bytes.

- hvutil: Processing Manual Mode request (*request*) for satellite node *node*.

    Informational message.

    Action:
    None required.

- hvutil: RMS is not running on *<targethost>*

    Printed when 'hvutil -A *targethost*' is called indicating that RMS is not running on the named host.

    Action:
    None required.

- hvutil: RMS is running on *<targethost>*

    Printed when 'hvutil -A *targethost*' is called indicating that RMS is running on the named host.

    Action:
    None required.

● hvutil: The resource *<resource>* does not have a detector
associated with it

   The resource *resource* does not have a detector.

   Action:
   Issue 'hvutil -N' on a resource that has a detector.

● hvutil: The resource *<resource>* is not a valid resource

   The resource *resource* is not a valid resource.

   Action:
   Issue 'hvutil -N' on a resource that has a detector and is part of the
   resource graph.

● hvutil: time period of detector must be an integer.

   If the detector time period specified as an argument with 'hvutil -t' is
   not a number, hvutil is aborted and exits with exit code 6.

   Action:
   Make sure that the detector time period is an integer.

● hvutil: Unable to open the notification file *<path>* due to
reason: *reason*

   hvutil was unable to open the file *path* because of *reason*.

   Action:
   Contact field support.

● Invalid delay.

   If the delay specified for sending a message using hvsend is a number
   less than zero, this message is printed.

   Action:
   Provide a valid value for the delay.

● It may take few seconds to do Debug Information collection.

   As the hvdump utility dumps out the information regarding the resource
   graph, it prints this message while it is collecting the information.

Action:
>   None required.

● localfile *filename* does not exist or is not an ordinary file

>   If the localfile specified as an argument to hvrcp does not exist or if it is
>   not a regular file, hvrcp exits with exit code 7.

Action:
>   Make sure that the localfile exists and is an ordinary file.

● Manual Mode request (*request*) failed for satellite node *node.*

>   Critical error.

Action:
>   Contact field support.

● Manual Mode request successfully processed for satellite
  node *node.*

>   Informational message.

Action:
>   None required.

● Modification file name is missing on the command line,
  usage: hvmod [−i] [−l] −f config_file.us | −E | −L | [−i]
  [−l] −c "modification directives"

>   The hvmod utility has been invoked in a way that does not conform to its
>   expected usage. The utility exits with exit code 2.

Action:
>   Follow the expected usage for the utility.

● Name of the modification file is too long.

>   If the length of the name of the modification file (specified as an argument
>   through the −f option) or the modification directives (specified via the −c
>   option) is longer than 113 characters, this message is printed and hvmod
>   exits with exit code 4.

Action:
>   Make sure that the arguments specified via −f and −c options are not too
>   long.

● NOTICE: User has been warned of 'hvshut -A' and has elected
to proceed.

   When the user invokes 'hvshut -A' and elects to proceed with the
   command, this message is printed.

   Action:
   None required.

● NOTICE: User has been warned of 'hvshut -f -a' and has
elected to proceed.

   When the user has elected to proceed with the command 'hvshut -f -
   a' this message is printed to confirm the choice.

   Action:
   None required.

● NOTICE: User has been warned of 'hvshut -L' and has elected
to proceed.

   When the user invokes 'hvshut -L', and then has elected to proceed
   with the command then, this message is printed to confirm that 'hvshut
   -L' is being invoked.

   Action:
   None required.

● RELIANT_LOG_PATH is not defined

   When the hvlogclean utility is invoked without the -d option, it refers to
   the environment variable RELIANT_LOG_PATH, to locate the
   hvloginit script. If the value of the variable cannot be found, this
   message is printed and the utility exits with exit code 6.

   Action:
   Make sure that the environment variable RELIANT_LOG_PATH has not
   been unset and is set to the appropriate value.

● RELIANT_PATH is not defined

   When the hvlogclean utility is invoked without the -d option, it refers to
   the environment variable RELIANT_PATH to locate the hvloginit script.
   If the value of the variable cannot be found, this message is printed and
   the utility exits with exit code 6.

Action:
> Make sure that the environment variable RELIANT_PATH is set to the appropriate value.

● Reset of RMS has been aborted.

> When the user invokes hvreset, the hvreset utility asks for a confirmation. If the answer is not yes then hvreset is aborted and this message is printed.

Action:
> None required.

● RMS environment failure: The following required variable is not defined in RMS environment:

> One of the environment variables required by RMS is missing in hvenv.

Action:
> Critical error. Check the environment definitions in the hvenv.* files to see if they have been corrupted.

● RMS has failed to start!
didn't find a valid entry in the RMS default configuration file "*configfilename*"

> This message appears when the RMS default configuration file exists but does not contain a valid reference to a configuration to run.

Action:
> Either place a default configuration file name in the RMS default configuration file or put the current configuration name in it that the user wants to start.

● RMS has failed to start!
hvcm has been invoked without specifying a configuration with the -c attribute, but with specifying other command line options. This may cause ambiguity and is therefore not possible. Please specify the entire commandline or use "hvcm" without further options to run the default configuration

> This message appears when the user tries to start RMS without the -c option and specifying other commandline options.

Action:

> When using `hvcm` with the `-c` option, '`-c` *configname*' should be the last arguments on the command line. Alternatively, to use with the default configuration, enter `hvcm` without any arguments to start RMS on the local node, and '`hvcm -a`' to start RMS aon all nodes.

● `RMS has failed to start!`
  `invalid entry in the RMS default configuration file "`*config-filename*`"`

> The user is not allowed to start RMS if the default configuration has invalid entry in the RMS default configuration file. The possible valid entries are 1. *configname* or 2. '`hvcm` *<options>* `-c` *<configname>*'. Refer to the `hvcm` man page for details on valid options in format 2.

Action:

> Remove all invalid entries in the RMS default configuration file. Refer the `hvcm` man page.

● `RMS has failed to start!`
  `multiple entries in the RMS default configuration file`
  `"`*configfilename*`"`

> The user is not allowed to start RMS if there are multiple entries in the default configuration file config.us.

Action:

> The user has to remove all the obscure entries in the RMS default configuration file and has to have only one valid configuration in it.

● `RMS has failed to start!`
  `RELIANT_HOSTNAME is not defined in the RMS environment`

> The environment variable `RELIANT_HOSTNAME` is not properly set.

Action:

> Ensure that the RMS environment variable `RELIANT_HOSTNAME` wasn't set erroneously to "" (null string) or explicitly unset in `hvenv.local`.

● RMS has failed to start!
  the number of arguments specified at the command line
  overrides the internal buffer of the RMS start utility

> This message appears when the number of arguments specified at the
> command line is more than the buffer capacity (= 30 command line
> arguments).

Action:
> Refer to the hvcm manual page for the correct syntax and usage.

● RMS has failed to start!
  the number of arguments specified at the RMS default config-
  uration file "*configfilename*" overrides the internal buffer of
  the RMS start utility

> This message appears when the user tries to start the RMS using the
> RMS default configuration file but unable to do so because the number
> of arguments specified in the RMS default configuration file overrides the
> internal buffer of the RMS start utility.

Action:
> Remove some of the unwanted arguments from the RMS default config-
> uration file. Check the man page for hvcm to get the required options to
> start RMS.

● RMS has failed to start!
  the options "-a" and "-s" are incompatible and may not be
  specified both

> This message appears when the user tries to start RMS uses the options
> -a and -s simultaneously.

Action:
> Check the man page for hvcm to get the format.

● rms is dead

> The hvrcp utility checks whether the RMS base monitor is alive every 10
> seconds, if it finds that it is not alive, it prints this message and exits with
> exit code 1.

Action:
> Get RMS running on the host.
> RMS on node *node* could not be shutdown with hvshut -A.

● `Root access required to start hvcm`

   To start RMS the user must have root access.

   Action:
   login as root and try `hvcm`.

● `Sending` *`data`* `to` *`resource`*`.`

   This message is printed when logging is turned on and *data* is being sent
   to the object *resource*.

   Action:
   None required.

● `Shutdown of RMS has been aborted.`

   When the user invokes '`hvshut -L`', the `hvshut` utility asks for a confir-
   mation, if the answer is `no` then '`hvshut -L`' is aborted and this message
   is printed.

   Action:
   None required.

● `Starting Reliant Monitor Services now`

   When RMS is starting up, this message is printed.

   Action:
   None required.

● `Starting RMS on remote host` *`host`* `now`

   This message will be printed when RMS is being started on the remote
   host *host*.

   Action:
   None required.

● `startup aborted per user request`

   When RMS is being started up with the "-c" option, if the configuration file
   specified is different from the entry in CONFIG.rms, RMS asks for confir-
   mation from the user, if he wants to activate the different configuration
   file. If the response is "no", then the this message is printed.

Action:

    None required.

● `The command '`*command*`' could not be executed`

    The execution of the command *command* failed.

Action:

    Check to see if the *command* is available.

● `The command '`*command*`' failed to reset uid information with errno '`*errno*`' — '`*errorreason*`'.`

    The execution of the command *command* failed trying to reset effective uid.

Action:

    Depends on the errno value.

● `The command '`*command*`' failed to set the effective uid information with errno '`*errno*`' — '`*errorreason*`'.`

    The execution of the command *command* failed trying to set the effective uid.

Action:

    Depends on the errno value.

● `The command '`*command*`' may not be executed on a node configured as a satellite node. Use Adaptive Services to perform the operation.`

    The execution of the command *command* on a satellite node is not supported.

Action:

    Use Adaptive Services to perform the operation.

● `The command '`*command*`' must be executed on a core node in a satellite configuration.`

    The execution of the command *command* should be performed on a core node in a satellite configuration.

Action:

    Execute the command from a core node in a satellite configuration.

● The configuration file "*nondefaultconfig*" has been specified as option argument of the -c option, but the Wizard Tools activated configuration is "*defaultconfig*" (see *defaultconfig*). The base monitor will not be started. The desired configuration file must be re-activated by using PCS Wizard activation command.

This message is shown when the user tries to start the RMS with a configuration different from the configuration present in the RMS default configuration file. The base monitor is not started, the user will need to either change the default configuration file by re-activating the configuration via the Wizard Tools hvw command or specify the proper option argument for the -c option.

Action:
The user should correct the default configuration by activating the specified configuration file using the Wizard Tools or specify the proper option argument to the -c option.

● The file '*filename*' could not be opened: *errormsg*

While performing a hvdump, if the file *filename* could not be opened because of *errormsg*, this message is printed and hvdump exits with exit code 8.

Action:
Take action based on the *errormsg*.

● The length of return message from BM is illegal (*actuallength* actual *expectedlength* expected).

When the hvassert utility expecting a return message from the base monitor receives a message of length *actuallength* when it is expecting a message of length *expectedlength*, this message is printed and hvassert exits with exit code 5.

Action:
Contact field support.

● The system call *systemcall* could not be executed: *errormsg*

While performing a hvdump, if the *systemcall* could not be executed because of *errormsg*, this message is printed and hvdump exits with exit code 7.

Action:
> Take action based on the *errormsg*.

● The user has invoked the hvcm command with the -a flag on a host where RMS is already running, sending request to start all remaining hosts.

> If hvcm is invoked with the -a flag, then RMS will be started on the other hosts in the cluster.

Action:
> None required.

● timed out! Most likely rms on the remote host is dead.

> While performing hvrcp, if the command times out because the base monitor on the local host has not received an acknowledgement from the base monitor on the remote host, the most probable reason is that the RMS on the remote host is dead.

Action:
> Make sure that the RMS on the remote host is running.

● Too many arguments, usage: hvmod -E

> The hvmod utility does not expect any arguments when invoked with the -E option. If arguments are supplied, hvmod exits with exit code 1.

Action:
> Make sure that 'hvmod -E' is not invoked with any arguments.

● Too many asserted objects, *maximum* is the max.

> Any attempt to assert on a number of objects that is greater than the *maximum* will cause this message to be printed.

Action:
> Make sure that the number of asserted objects is less than the max.

● Usage: hvassert [-h SysNode] [-q] -s resource_name
  resource_state | [-h SysNode] [-q] -w resource_name
  resource_state seconds | [-h SysNode] [-q] -d resource_name
  state_detail [seconds]

  If hvassert utility has been invoked in a way that does not conform to its
  expected usage. The utility exits with exit code 6.

  Action:
  Follow the usage specified above.

● Usage: hvcm [-V] [-a] [-s targethost] [-c config_file] [-m]
  [-h time] [-l level]

  The hvcm utility has been invoked in a way that does not conform to its
  expected usage.

  Action:
  See the hvcm man page for correct usage.

● Usage: hvconfig -l | -o config_file

  The hvconfig utility has been invoked in a way that does not conform to
  its expected usage. The utility exits with exit code 6.

  Action:
  Follow the expected usage for the utility.

● Usage: hvdisp {-a | -c | -h | -i | -l | -n | -S resource_name
  [-u | -c] | -z resource_name | -T resource_type [-u | -c] |
  -u | resource_name | ENV | ENVL} [-o out_file]

  The hvdisp utility has been invoked in a way that does not conform to its
  expected usage. The utility exits with exit code 6.

  Action:
  Follow the expected usage for the utility.

● Usage: hvdump {-g | -f out_file | -t wait_time}

  The hvdump utility has been invoked in a way that does not conform to its
  expected usage. The utility exits with exit code 6.

  Action:
  Follow the expected usage for the utility.

- Usage: hveject -s host

  The hveject utility has been invoked in a way that does not conform to
  its expected usage. The utility exits with exit code of 2 or 6 depending on
  one of the following conditions:
  – If an unknown option is used, the exit code is 2.
  – If the hveject utility is invoked directly without any options or
    arguments, the exit code is 6.

  Action:
  Follow the expected usage for the utility.

- Usage: hvjoin -s host

  The hvjoin utility has been invoked in a way that does not conform to its
  expected usage. The utility exits with exit code of 2 or 6 depending on
  one of the following conditions:
  – If an unknown option is used, the exit code is 2.
  – If the hveject utility is invoked directly without any options or
    arguments, the exit code is 6.

  Action:
  Follow the expected usage for the utility.

- Usage: hvlogclean [-d]

  The hvlogclean utility has been invoked in a way that does not conform
  to its expected usage. The utility exits with exit code 6.

  Action:
  Follow the expected usage for the utility.

- Usage: hvmod [-i] [-l] -f config_file.us | -E | -L | [-i]
  [-l] -c "modification directives"

  The hvmod utility has been invoked in a way that does not conform to its
  expected usage. The utility exits with exit code 6 in any of the following
  situations:
  – hvmod is invoked without any options.
  – hvmod is invoked with the -l or -i options but with arguments when
    none are expected.

  Action:
  Follow the expected usage for the utility.

● Usage: hvrcp localfile node:remotefile

This message is printed of either of these conditions:
– The number of arguments specified is not equal to 2.
– The second argument is not specified in the form *node*:*remotefile*.
hvrcp then exits with exit code 6.

Action:
Follow the intended usage of hvrcp as specified above.

● Usage: hvreset [−t timeout] userApplication

The hvreset utility has been invoked in a way that does not conform to its expected usage. The utility exits with exit code 2.

Action:
Follow the expected usage for the utility.

● Usage: hvsend { [−m message] [−s system] [−w waittime] dest_object | −f in_file [dest_object] }

The hvsend utility has been invoked in a way that does not conform to its expected usage.

Action:
Follow the intended usage of the utility.

● Usage: hvshut {−f | −L | −a | −l | −s SysNode | −A}

The hvshut utility has been invoked in a way that does not conform to its expected usage. The utility exits with the exit code 6.

Action:
Follow the usage specified above.

● Usage: hvswitch [−f] userApplication [SysNode] | −p userApplication

The hvswitch utility has been invoked in a way that does not conform to its expected usage. The utility exits with exit code 6.

Action:
Follow the intended usage of the utility.

● Usage: hvutil {-a | -d | -f | -c | -s} userApplication | {-t n | -N string } resource | -L{0|1} resource | {-o | -u} SysNode | -l level | -w | -W | -i {all | userApplication} | -r | -m {on|off|forceoff} userApplication | -M {on|off|forceoff} | {-C | -E} userApplication

> This message could appear in any one of the following situations:
> – 'hvutil -u' is invoked with more than 1 argument. Exit code 7.
> – hvutil is invoked without any options or arguments. Exit code 7.
> – hvutil is invoked with an illegal option. Exit code 7.
> – 'hvutil -i' is used without an argument. Exit code 13.
> – 'hvutil -r' is used with an argument. Exit code 14.
> – 'hvutil {-w | -W}' is used with an argument. Exit code 9.
> – 'hvutil -n' is invoked with NoConfirm as the only argument. Exit code 5.
> – 'hvutil {-m | -M}' is invoked with an argument other than on, off, or forceoff. Exit code 16.
> – 'hvutil -m' is invoked without an argument, or 'hvutil -M' is invoked with an argument. Exit code 16.

Action:
> Follow the intended usage of hvutil.

6 Appendix—Operating system error numbers

Some RMS error messages display the operating system error number, <*errno*>, that was returned when a process such as a detector or script failed. These error numbers may provide important clues in diagnosing the problem. This appendix summarizes the error numbers and their meanings for the Solaris and Linux operating systems.

6.1 Solaris error numbers

Error number	Reason for error
1	Not owner
2	No such file or directory
3	No such process
4	Interrupted system call
5	I/O error
6	No such device or address
7	Arg list too long
8	Exec format error
9	Bad file number
10	No child processes
11	Resource temporarily unavailable
12	Not enough space
13	Permission denied
14	Bad address
15	Block device required
16	Device busy
17	File exists
18	Cross-device link

Table 4: Solaris *errno* error numbers and their meanings

Error number	Reason for error
19	No such device
20	Not a directory
21	Is a directory
22	Invalid argument
23	File table overflow
24	Too many open files
25	Inappropriate ioctl for device
26	Text file busy
27	File too large
28	No space left on device
29	Illegal seek
30	Read-only file system
31	Too many links
32	Broken pipe
33	Argument out of domain
34	Result too large
35	No message of desired type
36	Identifier removed
37	Channel number out of range
38	Level 2 not synchronized
39	Level 3 halted
40	Level 3 reset
41	Link number out of range
42	Protocol driver not attached
43	No CSI structure available
44	Level 2 halted
45	Deadlock situation detected/avoided
46	No record locks available

Table 4: Solaris *errno* error numbers and their meanings

Error number	Reason for error
47	Operation canceled
48	Operation not supported
49	Disc quota exceeded
50	Bad exchange descriptor
51	Bad request descriptor
52	Message tables full
53	Anode table overflow
54	Bad request code
55	Invalid slot
56	File locking deadlock
57	Bad font file format
58	Owner of the lock died
59	Lock is not recoverable
60	Not a stream device
61	No data available
62	Timer expired
63	Out of stream resources
64	Machine is not on the network
65	Package not installed
66	Object is remote
67	Link has been severed
68	Advertise error
69	Srmount error
70	Communication error on send
71	Protocol error
72	Locked lock was unmapped
73	Facility is not active
74	Multihop attempted

Table 4: Solaris *errno* error numbers and their meanings

Error number	Reason for error
75	Error 75
76	Error 76
77	Not a data message
78	File name too long
79	Value too large for defined data type
80	Name not unique on network
81	File descriptor in bad state
82	Remote address changed
83	Can not access a needed shared library
84	Accessing a corrupted shared library
85	.lib section in a.out corrupted
86	Attempting to link in more shared libraries than system limit
87	Can not exec a shared library directly
88	Illegal byte sequence
89	Operation not applicable
90	Number of symbolic links encountered during path name traversal exceeds MAXSYMLINKS
91	Error 91
92	Error 92
93	Directory not empty
94	Too many users
95	Socket operation on non-socket
96	Destination address required
97	Message too long
98	Protocol wrong type for socket
99	Option not supported by protocol
100	Error 100
101	Error 101

Table 4: Solaris *errno* error numbers and their meanings

Error number	Reason for error
102	Error 102
103	Error 103
104	Error 104
105	Error 105
106	Error 106
107	Error 107
108	Error 108
109	Error 109
110	Error 110
111	Error 111
112	Error 112
113	Error 113
114	Error 114
115	Error 115
116	Error 116
117	Error 117
118	Error 118
119	Error 119
120	Protocol not supported
121	Socket type not supported
122	Operation not supported on transport endpoint
123	Protocol family not supported
124	Address family not supported by protocol family
125	Address already in use
126	Cannot assign requested address
127	Network is down
128	Network is unreachable
129	Network dropped connection because of reset

Table 4: Solaris *errno* error numbers and their meanings

Error number	Reason for error
130	Software caused connection abort
131	Connection reset by peer
132	No buffer space available
133	Transport endpoint is already connected
134	Transport endpoint is not connected
135	Structure needs cleaning
136	Error 136
137	Not a name file
138	Not available
139	Is a name file
140	Remote I/O error
141	Reserved for future use
142	Error 142
143	Cannot send after socket shutdown
144	Too many references: cannot splice
145	Connection timed out
146	Connection refused
147	Host is down
148	No route to host
149	Operation already in progress
150	Operation now in progress
151	Stale NFS file handle

Table 4: Solaris *errno* error numbers and their meanings

6.2 Linux error numbers

Error number	Reason for error
1	Operation not permitted
2	No such file or directory
3	No such process
4	Interrupted system call
5	Input/output error
6	No such device or address
7	Argument list too long
8	Exec format error
9	Bad file descriptor
10	No child processes
11	Resource temporarily unavailable
12	Cannot allocate memory
13	Permission denied
14	Bad address
15	Block device required
16	Device or resource busy
17	File exists
18	Invalid cross-device link
19	No such device
20	Not a directory
21	Is a directory
22	Invalid argument
23	Too many open files in system
24	Too many open files
25	Inappropriate ioctl for device
26	Text file busy
27	File too large

Table 5: Linux *errno* error numbers and their meanings

Error number	Reason for error
28	No space left on device
29	Illegal seek
30	Read-only file system
31	Too many links
32	Broken pipe
33	Numerical argument out of domain
34	Numerical result out of range
35	Resource deadlock avoided
36	File name too long
37	No locks available
38	Function not implemented
39	Directory not empty
40	Too many levels of symbolic links
41	Unknown error 41
42	No message of desired type
43	Identifier removed
44	Channel number out of range
45	Level 2 not synchronized
46	Level 3 halted
47	Level 3 reset
48	Link number out of range
49	Protocol driver not attached
50	No CSI structure available
51	Level 2 halted
52	Invalid exchange
53	Invalid request descriptor
54	Exchange full
55	No anode

Table 5: Linux *errno* error numbers and their meanings

Error number	Reason for error
56	Invalid request code
57	Invalid slot
58	Unknown error 58
59	Bad font file format
60	Device not a stream
61	No data available
62	Timer expired
63	Out of streams resources
64	Machine is not on the network
65	Package not installed
66	Object is remote
67	Link has been severed
68	Advertise error
69	Srmount error
70	Communication error on send
71	Protocol error
72	Multihop attempted
73	RFS specific error
74	Bad message
75	Value too large for defined data type
76	Name not unique on network
77	File descriptor in bad state
78	Remote address changed
79	Can not access a needed shared library
80	Accessing a corrupted shared library
81	.lib section in a.out corrupted
82	Attempting to link in too many shared libraries
83	Cannot exec a shared library directly

Table 5: Linux *errno* error numbers and their meanings

Error number	Reason for error
84	Invalid or incomplete multibyte or wide character
85	Interrupted system call should be restarted
86	Streams pipe error
87	Too many users
88	Socket operation on non-socket
89	Destination address required
90	Message too long
91	Protocol wrong type for socket
92	Protocol not available
93	Protocol not supported
94	Socket type not supported
95	Operation not supported
96	Protocol family not supported
97	Address family not supported by protocol
98	Address already in use
99	Cannot assign requested address
100	Network is down
101	Network is unreachable
102	Network dropped connection on reset
103	Software caused connection abort
104	Connection reset by peer
105	No buffer space available
106	Transport endpoint is already connected
107	Transport endpoint is not connected
108	Cannot send after transport endpoint shutdown
109	Too many references: cannot splice
110	Connection timed out
111	Connection refused

Table 5: Linux *errno* error numbers and their meanings

Error number	Reason for error
112	Host is down
113	No route to host
114	Operation already in progress
115	Operation now in progress
116	Stale NFS file handle
117	Structure needs cleaning
118	Not a XENIX named type file
119	No XENIX semaphores available
120	Is a named type file
121	Remote I/O error
122	Disk quota exceeded
123	No medium found
124	Wrong medium type

Table 5: Linux *errno* error numbers and their meanings

7   Appendix—Object types

The following alphabetical list describes all object types that are supplied with RMS and configured by PCS or the Wizard Tools.

- **andOp**

  Required attributes:
  - `HostName` (for direct children of a `userApplication` object)

  Object associated with its children by a logical *AND* operator. This object type is online if all children are online, and offline if all children are offline.

- **Controller**

  Required attributes:
  - `Resource`

  Object that allows a `userApplication` to control one or more `userApplication` objects. Not available with Adaptive Services.

- **ENV**

  Required attributes:
  - (none required)

  Object containing clusterwide (global) environment variables.

- **ENVL**

  Required attributes:
  - (none required)

  Object containing node-specific (local) environment variables.

- **gResource**

  Required attributes:
  - `rKind`
  - `rName`

  Custom (generic) object. Usually represents system resources such as file systems, network interfaces, or system processes.

- **orOp**

  Required attributes:
  – (none required)

  Object associated with its children by a logical *OR* operator. This object type is online if at least one child is online.

- **SysNode**

  Required attributes:
  – (none required)

  Represents nodes in the cluster; at least one required. Only `userApplication` objects are allowed as its children.

- **userApplication**

  Required attributes:
  – (none required)

  Represents an application to be monitored; at least one required. Must have one or more `SysNode` objects as its parents. For each `SysNode` parent, it must have one child `andOp` with its `HostName` attribute set to the name of the corresponding `SysNode`.

8 Appendix—Attributes

Some object types require specific attributes for RMS to monitor that object type. Some attributes can be modified through the user interface, while others are managed internally by PCS and the Wizard Tools. The following sections list all attributes along with their possible settings and default values.

8.1 Attributes available to the user

Attributes in this section can be changed using the PCS or Wizard Tools user interface.

- `AlternateIp`

  Possible Values: Any interconnect name
  Default: "" (empty)

  Valid for `SysNode` objects. Space-separated list that RMS uses as additional cluster interconnects if the interconnect assigned to the `SysNode` name becomes unavailable. All these interconnects must be found in the `/etc/hosts` database. By default, the configuration wizards assume the alternate interconnects to node *<nodename>* have names of the form *<nodename>*`rmsAI`*<nn>*, where *<nn>* is a two-digit, zero-filled number. This setting is restricted to very specific configurations and must never be used in a cluster with CF as interconnect.

- `AutoRecover`

  Possible Values: 0, 1
  Default: 0

  Valid for resource objects. If set to 1, executes the online script for an object if the object becomes faulted while in an `Online` state. If the object is able to return to the `Online` state, the fault is recovered.

- `AutoStartUp`

  Possible Values: 0, 1
  Default: 0

  Valid for `userApplication` objects. Automatically brings an application `Online` on the `SysNode` with the highest priority when RMS is started. Set to either 0 for no or 1 for yes.

You can override the `AutoStartUp` attribute for all `userApplication` objects by setting the `HV_AUTOSTARTUP` variable. See the description of `HV_AUTOSTARTUP` in the section "Local environment variables" on page 200.

● `AutoSwitchOver`

Possible Values: Valid string containing one or more of the following: `No`, `HostFailure`, `ResourceFailure`, `ShutDown`
Default: `No`

Valid for `userApplication` objects. Configures an application for automatic switchover if it becomes faulted. The values can be combined using the vertical bar ("|") character. The `No` value cannot be combined with any other value.

For backward compatibility, the numeric values 0 and 1 are accepted: 0 is equivalent to `No`, and 1 is equivalent to `HostFailure | Resource-Failure`.

● `ClusterExclusive`

Possible Values: 0, 1
Default: 0

Valid for resource objects. If set to 1, guarantees that the resource is `Online` on only one node in the cluster at any time. If set to 0, allows a resource to be `Online` on more than one node at a time.

The user can modify this attribute for a `cmdline` subapplication only. The configuration tools control this attribute for all other subapplications.

● `FaultScript`

Possible Values: Valid script (character)
Default: "" (empty)

Valid for all object types. Specifies a script to be run if the associated resource enters the `Faulted` state.

● `Halt`

Possible Values: 0, 1
Default: 0

Valid for `userApplication` objects. Eliminates a node if a double fault occurs and the `userApplication` can be switched to another node.

● `I_List`

Possible Values: Space-separated list of `SysNode` names
Default: "" (empty)

Valid for all `SysNode` objects. List of additional cluster interconnects that should be monitored by RMS. These interconnects are used only by customer applications and not by any PRIMECLUSTER products. All monitored interconnects must be found in the `/etc/hosts` database. In addition, all `SysNode` objects must have the same number of additional interconnects.

● `MaxControllers`

Possible Values: 0–512
Default: 512

Valid for `userApplication` objects. Upper limit of parent `userApplication` objects for the specified child application.

● `MonitorOnly`

Possible Values: 0, 1
Default: 0

Valid for resource objects. If set to 1, the state of the object is ignored by the parent when calculating the parent's state. Any parent should have at least one child for which `MonitorOnly` is not set.

● `OfflineScript`

Possible Values: Valid script (character)
Default: "" (empty)

Valid for all object types except `SysNode` objects. Specifies the script to be run to bring the associated resource to the `Offline` state.

● `OnlinePriority`

Possible Values: 0, 1
Default: 0

Valid for `userApplication` objects. Allows RMS to start the application on the node where it was last online when the entire cluster was brought down and then restarted. If set to 0 or not set (the default), the application comes online on the node with the highest priority in the attribute `PriorityList`. If

set to 1, the application comes online on the node where it was last online. In case of `AutoStartUp` or a priority switch, this last-online node has the highest priority, regardless of its position in the priority list.

RMS keeps track of where the application was last online by means of times-tamps. The node which has the latest timestamp for an application is the node on which the application will go online. Different cluster nodes should be in time-synchronization with each other, but this is not always the case. Since RMS does not provide a mechanism for ensuring time-synchroni-zation between the nodes in the cluster, this responsibility is left to the system administrator. If RMS detects a severe time-discrepancy between the nodes in the cluster, an ERROR message is printed to the switchlog.

NTPD may be used to establish consistent time across the nodes in the cluster. Refer to the manual page for `xntpd` for more information.

The `OnlinePriority` persistent state information will be cleared if RMS is restarted with the last online node removed from the configuration.

● **OnlineScript**

Possible Values: Valid script (character)
Default: "" (empty)

Valid for all objects except `SysNode` objects. Specifies the script to bring the associated resource to the `Online` state.

● **PartialCluster**

Possible Values: 0, 1
Default: 0

Valid for `userApplication` objects. Specifies if an application can negotiate online requests.

If set to 0, then the application can negotiate its online request only when all nodes where it can possibly run are online.

If set to 1, then the application can negotiate its online request within the current set of online nodes, even if some other nodes (including the appli-cation's primary node) are offline or faulted.

Note that a local `userApplication` that has its `PartialCluster` attribute set will not be affected by startup timeouts from remote nodes: the application can still go online on the local node. See the description of **HV_AUTOSTARTUP_WAIT** in the section "Global environment variables" on page 196.

● `PersistentFault`

Possible Values: 0, 1
Default: 0

Valid for `userApplication` objects. If set to 1, the application maintains a `Faulted` state across an RMS shutdown and restart. The application returns to the `Faulted` state if it was `Faulted` before, unless the fault is explicitly cleared by either 'hvutil -c' or 'hvswitch -f', or if RMS is restarted with the `Faulted SysNode` removed from the configuration.

● `PostOfflineScript`

Possible Values: Valid script (character)
Default: "" (empty)

Valid for all objects except `SysNode` objects. Specifies the script to be run after the state of the associated resource changes to `Offline`.

● `PostOnlineScript`

Possible Values: Valid script (character)
Default: "" (empty)

Valid for all objects except `SysNode` objects. Specifies the script to be run after the state of the associated resource changes to `Online`.

● `PreOfflineScript`

Possible Values: Valid script (character)
Default: "" (empty)

Valid for all objects except `SysNode` objects. Specifies the script to run before the object is taken to the `Offline` state.

● `PreOnlineScript`

Possible Values: Valid script (character)
Default: "" (empty)

Valid for all objects except `SysNode` objects. Specifies the script to be run before the associated resource is taken to the `Online` state.

● `PreserveState`

Possible Values: 0, 1
Default: 0

Valid for `userApplication` objects. Specifies that resources are not to be taken `Offline` after a fault. Ignored if `AutoSwitchOver` is not set to `No`.

● `PriorityList`

Possible Values: Valid list of `SysNode` names (character)
Default: "" (empty)

Valid for `userApplication` objects. Contains a list of `SysNode` objects where the application can come `Online`. The order in the list determines the next node to which the application is switched during a priority switchover, ordering a switchover after a `Fault`. The list is processed circularly.

The user specifies this attribute indirectly when selecting the nodes for an application. RMS uses the order in which the nodes were selected and creates `PriorityList` automatically. The user can change the `PriorityList` by adding individual nodes from the list in the desired order, rather than automatically selecting the entire list.

For applications controlled by a controller, the order of nodes in `PriorityList` is ignored. However, each child application must be able to run on the nodes specified for the controller object.

● `ScriptTimeout`

Possible Values: 0–*MAXINT* (in seconds) or valid string of the form
"*timeout_value*[:[*offline_value*][:*online_value*]]"
Default: 300

Valid for all object types. Specifies the timeout value for all scripts associated with that object in the configuration file. RMS sends a kill signal to the script if the timeout expires.

Use the string format to specify individual timeout values of *offline_value* for `OfflineScript` and *online_value* for `OnlineScript`.

● **ShutdownPriority**

Possible Values: 0–*MAXINT*
Default: 0

Valid for `userApplication` objects. `ShutdownPriority` assigns a weight factor to the application for use by the Shutdown Facility.

When interconnect failures and the resulting concurrent node elimination requests occur, SF calculates the shutdown priority of each subcluster as the sum of the subcluster's SF node weights plus the RMS `ShutdownPriority` of all online application objects in the subcluster. The optimal subcluster is defined as the fully connected subcluster with the highest weight.

● **StandbyCapable**

Possible Values: 0, 1
Default: 0

Valid for resource objects. If set to 1, the object performs standby processing on all nodes where the parent application is supposed to be `Offline`.

The user can modify this attribute for a `cmdline` subapplication only. The configuration tools control this attribute for all other subapplications.

● **StandbyTransitions**

Possible Values: `StartUp, SwitchRequest, ClearFaultRequest` or any combination joined by vertical bars (|)
Default: "" (empty)

Valid for `userApplication` objects. The value specifies when the application will receive a standby request:

– *StartUp*—at startup, unless the application is already online or unless it is forced to go online due to the `AutoStartUp` attribute.

– *SwitchRequest*—after application switchover, if the application was online before the switchover.

– *ClearFaultRequest*—after a faulted state is cleared with 'hvutil -c'.

● `WarningScript`

Possible Values: Valid script (character)
Default: "" (empty)

Valid for resource objects with detector. Specifies the script to be run after
the posted state of the associated resource changes to `Warning`.

8.2   Attributes managed by configuration wizards

Attributes in this section are managed internally by the configuration wizards or by RMS at runtime.

- `Affiliation`

  Possible Values: Any string
  Default: "" (empty)

  Valid for resource objects. Used for display purposes in the user interface—no functional meaning within RMS.

- `AutoRecoverCleanup`

  Possible Values: 0, 1
  Default: 1

  Valid for `controller` objects. If set to 1, and `AutoRecover` is 1, then a faulted child application is requested to go `Offline` before recovering. If set to 0 and `AutoRecover` is 1, then a faulted child application recovers without going `Offline`.

- `Class`

  Possible Values: any string
  Default: Default type defined in the chapter "Appendix—Object types" on page 179.

  Valid for all objects except `SysNode`. Describes the class of the resource object. Used by other programs for various purposes (for example, SNMP agent). This value is supplied by the configuration wizards.

- `Comment`

  Possible Values: any string
  Default: "" (empty)

  Valid for all objects. Used for documentation in the configuration file—no functional meaning within RMS.

●   `ControlledShutdown`

Possible Values: 0, 1
Default: 0

Valid for controlled `userApplication` objects. If set to 1, RMS does not
send an `Offline` request to this application because an explicit request will
be generated by a parent application during its offline processing.

●   `ControlledSwitch`

Possible Values: 0, 1
Default: 0

Valid for controlled `userApplication` objects. If set to 1, the application is
the child of a controller.

●   `DetectorStartScript`

Possible Values: Any valid detector start script
Default: "" (empty)

Valid for resource object with detector. Specify the detector start command
directly in the *<configname>*`.us` file.

●   `HostName`

Possible Values: Any `SysNode` name
Default: "" (empty)

Must be set only in the first-level `andOp` children of a `userApplication`
object. Each of these `andOp` objects associates its parent application with
the `SysNode` specified in its `HostName` attribute; the child `andOp` objects also
determine the priority of the application's nodes.

●   `IgnoreStandbyRequest`

Possible Values: 0, 1
Default: 1

Valid for `controller` objects. If set to 1, then neither `PreOnline` nor
`Online` requests during standby processing will be propagated to the child
application. If 0, then requests will be propagated. If the controller is not
standby capable, then `IgnoreStandbyRequest` must be set to 1.

- `LastDetectorReport`

  Possible Values: `Online`, `Offline`, `Faulted`, `Standby`
  Default: (none)

  Valid for resource objects with detector. This attribute contains the most recent detector report for the object. The value may be displayed in the Cluster Admin GUI; the possible values depend on the type of resource the object represents.

- `LieOffline`

  Possible Values: `0`, `1`
  Default: `1`

  Valid for all resource objects. If set to 1, allows the resource to remain `Online` during `Offline` processing.

- `NoDisplay`

  Possible Values: `0`, `1`
  Default: `0`

  Valid for all object types. If set to 1, specifies that the resource should not be displayed when `hvdisp` is active. Can be overridden by '`hvdisp -S`'.

- `NullDetector`

  Possible Values: `on`, `off`
  Default: `off`

  Valid for resource objects with detector. Used to disable a detector at runtime by setting `NullDetector` to `on`. This attribute is for the use with dynamic reconfiguration only. `NullDetector` must never be set hard-coded to `on` in the RMS configuration file.

- `OfflineDoneScript`

  Possible Values: Valid script (character)
  Default: "" (empty)

  Valid for `userApplication` objects. The last script run after the application has completed offline processing.

- `OnlineTimeout`

    Possible Values: 0–*MAXINT*
    Default: 0

    Valid for `controller` objects. Specifies the time (in seconds) allowed for a controller not to react while a child application leaves the `Online` state

- `PreCheckScript`

    Possible Values: Valid script (character)
    Default: "" (empty)

    Valid for `userApplication` objects. Specifies the script to be forked as the first action during `Online` or `Standby` processing. If the script returns with a zero exit code, processing proceeds. If the script returns with an exit code other than zero, processing is not performed and an appropriate warning is logged to the `switchlog` file.

- `Resource`

    Possible Values: Valid name (character)
    Default: "" (empty)

    Valid for `controller` objects. One or more names of child applications, separated by spaces and/or tabs.

- `rKind`

    Possible Values: 0–2047
    Default: none

    Valid for `gResource` objects. Specifies the kind of detector for the object.

- `rName`

    Possible Values: Valid string (character)
    Default: none

    Valid for `gResource` objects. Specifies a string to be forwarded to the generic detector.

- `SplitRequest`

    Possible Values: 0, 1
    Default: 0

Valid for `controller` objects. If set to 1, then `PreOffline` and `Preonline` requests will be propagated to child applications separately from the `Offline` and `Online` requests. If 0, then separate `PreOffline` or `PreOnline` requests will not be issued for the child applications.

9 Appendix—Environment variables

This appendix provides a complete list of the environment variables used by RMS, grouped into the following types:

- "Global environment variables" on page 196

- "Local environment variables" on page 200

- "Script execution environment variables" on page 204

9.1 Setting environment variables

⚠️ **Caution**

Do not explicitly set RMS environment variables in the user environment. Doing so can cause RMS to lose environment variables settings.

ℹ️ Do not change the `hvenv` configuration file. Changes to your configuration's environment variables should be confined to the `hvenv.local` file.

The values of environment variables are specified as export directives in the `hvenv.local` file. To adjust a variable's setting, you would open `hvenv.local` with a text editor of your choice and modify (or add) the appropriate line.

A typical export directive would appear as follows:

```
export SCRIPTS_TIME_OUT=200
```

When RMS starts, it reads the values of environment variables from `hvenv` and `hvenv.local` and initializes the `ENV` and `ENVL` objects respectively. No further reference is made to these two configuration files while RMS is running. Therefore, any changes you make to `hvenv.local` will not take effect until the next time RMS starts up.

Values in the `ENVL` (local) object override values in the `ENV` (global) object. If a global variable setting appears in the `hvenv.local` file, it will override the corresponding setting in the `hvenv` file. However, if you adjust a global variable in the `hvenv.local` file on one node, you must make the same adjustment to `hvenv.local` on every other node in the cluster. Global variable settings must agree clusterwide.

While RMS is running, you can display the environment variables with the `hvdisp` command, which does not require root privilege:

● `hvdisp ENV`

● `hvdisp ENVL`

9.2 Global environment variables

> **i** Global variable settings (`ENV`) are included in the configurations checksum that is common to the cluster. The checksum is verified on each node during startup of the base monitor. RMS will fail to start if it detects a checksum difference between the values on any two nodes.

> **i** The default values of the environment variables are found in *<RELIANT_PATH>*`/bin/hvenv`. They can be redefined in the `hvenv.local` configuration file.

The following list describes the global environment variables for RMS:

● `HV_AUTOSTARTUP_IGNORE`

Possible values: List of RMS cluster nodes. The list of RMS cluster nodes must be the names of the `SysNode`s as found in the RMS configuration file. The list of nodes cannot include the CF name.
Default: "" (empty)

List of cluster nodes that RMS ignores when it starts. This environment variable is not set by default. A user application will begin its automatic startup processing if the `AutoStartUp` attribute is set and when all cluster nodes defined in the user application have reported `Online`. If a cluster node appears in this list, automatic startup processing will begin even if this node has not yet reported the `Online` state.

Use this environment variable if one or more cluster nodes need to be taken out of the cluster for an extended period and RMS will continue to use the configuration file that specifies the removed cluster nodes. In this case, specifying the unavailable cluster nodes in this environment variable ensures that all user applications are automatically brought online even if the unavailable cluster nodes do not report `Online`.

⚠️ **Caution**

If the HV_AUTOSTARTUP_IGNORE environment variable is used, ensure that it is correctly defined on all cluster nodes and that it is always kept up-to-date. When a node is brought back into the cluster, remove it from this environment variable. If this does not occur, data loss could occur because RMS will ignore this node during the startup procedure and will not check whether the application is already running on the nodes specified in this list. It is the system administrator's responsibility to keep this list up-to-date if it is used.

● **HV_AUTOSTARTUP_WAIT**

Possible values: 0–*MAXINT*
Default: 60 (seconds)

Defines the period (in seconds) that RMS waits for cluster nodes to report Online when RMS is started. If this period expires and not all cluster nodes are online, a switchlog message indicates the cluster nodes that have not reported Online and why the user application(s) cannot be started automatically.

Note that **HV_AUTOSTARTUP_WAIT** timeouts from remote nodes will not affect a local userApplication that has its PartialCluster attribute set: the application can still go online on the local node. See the description of the PartialCluster attribute in the section "Attributes available to the user" on page 181.

ℹ️ This attribute generates a warning message only. AutoStartUp will proceed even if the specified period has expired.

● **HV_CHECKSUM_INTERVAL**

Possible values: 0–*MAXINT*
Default: 120 (seconds)

Interval in seconds for which the RMS base monitor waits for each Online node to verify that its checksum is the same as the local checksum.

If checksums are confirmed within this interval, then RMS on the local node continues its operations as usual. However, if a checksum from a remote node is not confirmed, or if it is confirmed to be different, then the local monitor shuts down if it has been started less than HV_CHECKSUM_INTERVAL seconds before.

Also, if a checksum from a remote node is not confirmed, or if the checksum is confirmed to be different, then the local monitor considers the remote node as `Offline` if that local monitor has been started more than `HV_CHECKSUM_INTERVAL` seconds before.

● `HV_LOG_ACTION_THRESHOLD`

   Possible values: 0–100
   Default: 98

   Defines the behavior of `hvlogcontrol`. If the used space on the log disk is larger or equal to this threshold, all subdirectories below log will be removed. Furthermore, if `HV_LOG_ACTION` is set to `on` and all subdirectories have already been removed, the actual log files will be removed too Refer to the section "Local environment variables" on page 200 for more information on `HV_LOG_ACTION`.

● `HV_LOG_WARN_THRESHOLD`

   Possible values: 0–100
   Default: 95

   Defines the behavior of `hvlogcontrol`. If the used space on the file system containing the RMS log disk is larger or equal to this threshold value, the `hvlogcontrol` script issues a warning to the user regarding the large amount of log files.

● `HV_LOH_INTERVAL`

   Possible values: 0–*MAXINT*
   Default: 30

   Minimum difference in seconds when comparing timestamps to determine the last online host for an application. The last online host (LOH) specifies the host where the `userApplication` was online most recently. It is determined if the `OnlinePriority` attribute is set.

   If the LOH timestamp entries of the `userApplication` on two hosts differ by less than this time interval, RMS does not perform `AutoStartUp` and does not allow priority switches. Instead, it sends a message to the console and waits for operator intervention.

   When adjusting this variable, the quality of the time synchronization in the cluster must be taken into account. The value must be larger than any possible random time difference between the cluster hosts.

- HV_WAIT_CONFIG

  Possible values: 0–*MAXINT*
  Default: 120 (seconds)

  Interval in seconds during which RMS waits to receive a configuration from an online host if RMS starts up as 'hvcm -C'. If the configuration is not received within *HV_WAIT_CONFIG* seconds, the local monitor will attempt to run with the configuration file specified in *RELIANT_BUILD_PATH*. If such a file does not exist, the local monitor will continue to run with the minimal configuration; in this case it is possible for it to join an already running RMS cluster via hvjoin.

- RELIANT_LOG_LIFE

  Possible values: Any number of days
  Default: 7 (days)

  Specifies the number of days that RMS logging information is retained. Every time RMS starts, the system creates a directory that is named on the basis of when RMS was last started, and which contains all constituent log files. All RMS log files are preserved in this manner. All log files which are older than the number of days specified in this variable are deleted by a cron job.

- RELIANT_LOG_PATH

  Possible values: Any valid path
  Default: /var/opt/SMAWRrms/log

  Specifies the directory where all RMS, PCS, and Wizard Tools log files are stored.

- RELIANT_PATH

  Possible values: Any valid path
  Default: /opt/SMAW/SMAWRrms

  Specifies the root directory of the RMS directory hierarchy. Users do not normally need to change the default setting.

- RELIANT_SHUT_MIN_WAIT

  Possible values: 0–*MAXINT*
  Default: 900 (seconds)

Defines the period (in seconds) that the command `hvshut` waits before timing out and generating an error message. This value should be no less than the maximum time required by any application in the configuration to go offline on any node in the cluster.

If this value is too low, RMS terminates ungracefully: all running scripts are terminated immediately, and some resources under control of RMS will be left in a arbitrary state. These resources must be manually shut down before RMS can be restarted.

The default value will be adequate for some configurations, but each configuration must be considered individually. Long delays in offline processing may be caused by recurring issues such as large numbers of nodes or resources, or slow network connections or hardware. We recommend that you obtain the advice of an expert who is familiar with the applications and resources in your cluster.

If expert advice is unavailable, you can still estimate a reasonable value for `RELIANT_SHUT_MIN_WAIT`. Temporarily set the variable to a large value (e.g., 4000), run a series of tests that simulate production conditions, and then use the worst-case offline processing time plus a safety factor (e.g., 10%).

> **i** Due to the serious effects, you should diagnose the cause of an offline processing timeout before making another attempt to shut down RMS automatically.

9.3   Local environment variables

Local environment variable settings can vary from node to node. The following list describes the local environment variables for RMS:

- `HV_AUTOSTARTUP`

  Possible values: 0, 1
  Default: 1 (normal processing of `AutoStartUp` attribute)

  Controls the action of the `AutoStartUp` attribute for all `userApplication` objects on the local node. If set to 1 (the default value) the automatic startup of each `userApplication` is determined by its `AutoStartUp` attribute (see the section "Attributes available to the user" on page 181). If set to 0, the `AutoStartUp` attribute is ignored and no automatic startup occurs.

HV_AUTOSTARTUP can be set in the Cluster Admin *Tools* menu or by using the hvsetenv command; in either case, the change does not take effect until the next RMS startup.

● **HV_CONNECT_TIMEOUT**

Possible values: 0–*MAXINT*
Default: 0 (seconds). Users do not normally need to change the default setting.

The maximum time (in seconds) that the heartbeat from a node is not received before the base monitor assumes the connection to that node has been lost.

● **HV_LOG_ACTION**

Possible values: on, off
Default: off

Determines if the current log files in the directory RELIANT_LOG_PATH are deleted if the used space on the file system is larger or equal to HV_LOG_ACTION_THRESHOLD.

● **HV_MAX_HVDISP_FILE_SIZE**

Possible values: 0–*MAXINT*
Default: 20,000,000 (bytes)

Prevents the unlimited growth of the temporary file that RMS uses to supply hvdisp with configuration data and subsequent configuration and state changes. The value of this variable is the maximum size in bytes of the temporary file *<RELIANT_PATH>*/locks/.rms.*<process id of the hvdisp process>*.

● **HV_MAXPROC**

Possible values: 0–*fork limit*
Default: 30

Defines the maximum number of scripts RMS can have forked at any time. The default (30) is sufficient in most cases.

● `HV_RCSTART`

Possible values: 0, 1
Default: 1 (start RMS in the `rc` script)

Determines if RMS is started in the `rc` script. If set to 1 (the default value), RMS is started automatically at system boot time. If set to 0, RMS must be started manually. `HV_RCSTART` can be set in the Cluster Admin *Tools* menu or by using the `hvsetenv` command. (Prerequisite for `rc` start: `CONFIG.rms` exists and contains a valid entry.)

● `HV_REALTIME_PRIORITY`

Possible values: 0–99
Default: 50

Defines the real time priority for the RMS base monitor and its detectors. Caution should be used when adjusting this variable. High settings can prevent other OS real-time processes from getting their processor time slice. Low settings can prevent the RMS base monitor from reacting to detector reports and from performing requests from command line utilities.

● `HV_SCRIPTS_DEBUG`

Possible values: 0, 1
Default: 0

Controls debugging output from RMS scripts. If this variable is set to 1, each script writes detailed information about the commands that are executed to the RMS `switchlog` file. The type of information logged may vary according to the script. This setting applies only to those scripts provided with PRIME-CLUSTER products. To disable script debug message logging, delete the `HV_SCRIPTS_DEBUG` entry or set `HV_SCRIPTS_DEBUG=0` in `hvenv.local`.

● `HV_SYSLOG_USE`

Possible values: 0, 1
Default: 1 (in `hvenv`)

Controls output to the system log from the RMS base monitor. RMS always records RMS ERROR, FATAL ERROR, WARNING, and NOTICE messages in the RMS `switchlog` file. By default, these messages are duplicated in the system log file `/var/adm/messages` (Solaris) or `/var/log/messages` (Linux). To disable RMS messages in the system log, set `HV_SYSLOG_USE=0` in `hvenv.local`.

● **RELIANT_HOSTNAME**

Possible values: valid name
*Default: <nodename>*RMS

The name of the local node in the RMS cluster. The default value of this variable is the node name with an RMS suffix (for example: fuji2RMS), as generated by the following command:

```
export RELIANT_HOSTNAME=`cftool -l 2>/dev/null | tail -1 |
cut -f1 -d" "`RMS
```

```
export RELIANT_HOSTNAME=`uname -n`RMS
```

If this preset value is not suitable, it must be modified accordingly on all nodes in the cluster.

The specified cluster node name must correspond to the SysNode name in the *<configname>*.us configuration file. The node name determines the IP address that RMS uses for establishing contact with this node. See the *PRIMECLUSTER for mySAP Business Suite Installation Guide (Linux)* for instructions on how to modify the /etc/hosts database for RMS.

● **RELIANT_INITSCRIPT**

Possible values: any executable
Default: <RELIANT_PATH>/bin/InitScript

Specifies an initialization script to be run by RMS when the system is started. This script is run before any other processes are activated. It is a global script that is run once on every cluster node on which it is defined.

● **RELIANT_STARTUP_PATH**

Possible values: any valid path
Default: <RELIANT_PATH>/build

Defines where RMS searches at start time for the configuration files.

● `SCRIPTS_TIME_OUT`

   Possible values: 0–*MAXINT*
   Default: 300 (seconds)

   Specifies the global period (in seconds) within which all RMS scripts must be terminated. If a specific script cannot be terminated within the defined period, it is assumed to have failed and RMS begins appropriate processing for a script failure.

   If this value is too low, error conditions will be produced unnecessarily, and it may not be possible for the applications to go online or offline. An excessively high value is unsuitable because RMS will wait for this period to expire before assuming that the script has failed.

   In case the global setting is not appropriate for all objects monitored by RMS, this global value can be overridden by an object-specific setting of the `ScriptTimeout` attribute.

9.4   Script execution environment variables

The variables in this section are set by the RMS base monitor when it executes an object's script. These exist only in the script's environment and only for the duration of the script execution. Since these variables are explicitly set, they have no default values.

● `HV_APPLICATION`

   Possible values: any `userApplication` name

   Name of the `userApplication` object at the top of the sub-tree that contains the current object.

● `HV_AUTORECOVER`

   Possible values: 0, 1

   If set to 1, the script was initiated due to an `AutoRecover` attempt.

● `HV_FORCED_REQUEST`

   Possible values: 0, 1

   If set to 1, the script is currently processing a forced request.

● `HV_LAST_DET_REPORT`

   Possible values: one of `Online`, `Offline`, `Faulted`, `NoReport`

   Last detector report for the current object.

● `HV_OFFLINE_REASON`

   Possible values: one of `DEACT`, `SWITCH`, `FAULT`, `STOP`

   Reason for ongoing offline processing:
   `DEACT`: **deact request** (‘`hvutil -d`’)
   `SWITCH`: **manual switchover** (‘`hvswitch`’)
   `FAULT`: **follow-up processing after a previous resource failure**
   `STOP`: `userApplication` **is stopped** (‘`hvutil -f`’, ‘`hvutil -c`’, ‘`hvshut`’).

● `HV_NODENAME`

   Possible values: any object name

   Name of current object.

● `HV_SCRIPT_TYPE`

   Possible values: one of `PreCheckScript`, `PreOnlineScript`,
   `OnlineScript`, `PostOnlineScript`, `PreOfflineScript`,
   `OfflineScript`, `PostOfflineScript`, `OfflineDoneScript`,
   `FaultScript`

   Script type.

● `NODE_SCRIPTS_TIME_OUT`

   Possible values: 0–*MAXINT*

   Timeout value for the current object and script type.

10 Appendix—List of manual pages

This appendix lists the online manual pages for CCBR, CF, CFS, CIP, Monitoring Agent, PAS, PCS, RCVM, Resource Database, RMS, RMS Wizards, SCON, SF, SIS, and Web-Based Admin View.

To display a manual page, type the following command:

$ `man` *man_page_name*

10.1 CCBR

System administration

cfbackup
> save the cluster configuration information for a PRIMECLUSTER node

cfrestore
> restore saved cluster configuration formation on a PRIMECLUSTER node

10.2 CF

System administration

cfconfig
> configure or unconfigure a node for a PRIMECLUSTER cluster

cfset
> apply or modify `/etc/default/cluster.config` entries into the CF module

cfrecon
> dynamically reconfigure the cluster interconnects used by a node

cftool
> print node communications status for a node or the cluster

rcqconfig
> configures or reports cluster quorum settings

10.3   CFS

`fsck_rcfs`
> file system consistency check and interactive repair

`mount_rcfs`
> mount RCFS file systems

`rcfs_fumount`
> force unmount RCFS mounted file system

`rcfs_list`
> list status of RCFS mounted file systems

`rcfs_switch`
> manual switchover or failover of a RCFS file system

`ngadmin`
> node group administration utility

`cfsmntd`
> cfs mount daemon for RCFS

10.4   CIP

System administration

`cipconfig`
> start or stop CIP 2.0

`ciptool`
> retrieve CIP information about local and remote nodes in the cluster

File format

`cip.cf`
> CIP configuration file format

10.5 Monitoring Agent

System administration

clrcimonctl

> Start, stop or restart of the RCI monitoring agent daemon, and display of daemon presence

clrccumonctl

> Start, stop or restart of the console monitoring agent daemon, and display of daemon presence

10.6 PAS

System administration

mipcstat

> MIPC statistics

clmstat

> CLM statistics

10.7 PCS

System administration

pcstool

> Modifies PCS configurations from the command line

pcscui

> Character-based interface for PCS

pcs_reinstall

> Utility for re-integrating PCS with dependent products

10.8 RCVM

RCVM is not available in all markets.

System administration

dkconfig
> virtual disk configuration utility

dkmigrate
> virtual disk migration utility

vdisk
> virtual disk driver

dkmirror
> mirror disk administrative utility

File format

dktab
> virtual disk configuration file

10.9 Resource Database

| i | To display a Resource Database manual page, add /etc/opt/FJSVcluster/man to the environment variable MANPATH. |

System administration

clautoconfig
> execute of the automatic resource registration

clbackuprdb
> save the resource database

clexec
> execute the remote command

cldeldevice
> delete resource registered by automatic resource registration

clinitreset
> reset the resource database

clrestorerdb
      restore the resource database

clsetparam
      display and change the resource database operational environment

clsetup
      set up the resource database

clstartrsc
      resource activation

clstoprsc
      resource deactivation

clsyncfile
      distribute a file between cluster nodes

User command

clgettree
      display the tree information of the resource database

10.10 RMS

System administration

hvassert
      assert (test for) an RMS resource state

hvattr
      make cluster-wide attribute changes at runtime from a single node
      (installed with PCS or the Wizard Tools)

hvcm
      start the RMS configuration monitor

hvconfig
      display or save the RMS configuration file

hvdisp
      display RMS resource information

hvdist
      distribute RMS configuration files

`hvdump`
    collect debugging information about RMS

`hvgdmake`
    compile an RMS custom detector

`hvlogclean`
    clean RMS log files

`hvrclev`
    change default RMS start run level

`hvreset`
    reinitialize the graph of an RMS user application (for use by experts in
    test conditions only—not for use on production clusters)

`hvsetenv`
    controls automatic startup of RMS or all user applications on the local
    host

`hvshut`
    shut down RMS

`hvswitch`
    switch control of an RMS user application resource to another node

`hvthrottle`
    prevent multiple RMS scripts from running simultaneously

`hvutil`
    manipulate availability of an RMS resource

File formats

`hvenv.local`
    RMS local environment configuration file

10.11 RMS Wizards

RMS Wizard Tools and RMS Wizard Kit
    RMS Wizards are documented as HTML pages in the `SMAWRhv-do`
    package on the CD-ROM. After installing this package, the documen-
    tation is available in the following directory:

    <RELIANT_PATH>/`htdocs.solaris/wizards.en` (Solaris)

<RELIANT_PATH>/htdocs.linux/wizards.en (Linux)

The default value of *<RELIANT_PATH>* is /opt/SMAW/SMAWRrms/.

10.12 SCON

scon
> start the cluster console software

10.13 SF

System administration

rcsd
> shutdown daemon for the Shutdown Facility

sdtool
> interface tool for the shutdown daemon

File formats

rcsd.cfg
> configuration file for the Shutdown Daemon

SA_blade.cfg
> configuration file for FSC server blade Shutdown Agent

SA_rccu.cfg
> configuration file for RCCU Shutdown Agent

SA_rps.cfg
> configuration file for Remote Power Switch Shutdown Agent

SA_rsb.cfg
> configuration file for RemoteView Services Board Shutdown Agent

SA_scon.cfg
> configuration file for SCON Shutdown Agent

SA_pprci.cfg
> configuration file for RCI Shutdown Agent (PRIMEPOWER only)

SA_sspint.cfg
> configuration file for Sun E10000 Shutdown Agent

SA_sunF.cfg
> configuration file for sunF system controller Shutdown Agent

SA_wtinps.cfg
> configuration file for WTI NPS Shutdown Agent

10.14 SIS

System administration

dtcpadmin
> start the SIS administration utility

dtcpd
> start the SIS daemon for configuring VIPs

dtcpstat
> status information about SIS

10.15 Web-Based Admin View

System administration

fjsvwvbs
> stop Web-Based Admin View

fjsvwvcnf
> start, stop, or restart the web server for Web-Based Admin View

wvCntl
> start, stop, or get debugging information for Web-Based Admin View

wvGetparam
> display Web-Based Admin View's environment variable

wvSetparam
> set Web-Based Admin View environment variable

wvstat
> display the operating status of Web-Based Admin View

Glossary

Items in this glossary that apply to specific PRIMECLUSTER products are indicated with the following notation:

- (CF)—Cluster Foundation

- (PCS)—PRIMECLUSTER Configuration Services

- (RMS)—Reliant Monitor Services

- (RCVM)—Volume Manager (not available in all markets)

- (SIS)—Scalable Internet Services

Some of these products may not be installed on your cluster. See your PRIME-CLUSTER sales representative for more information.

AC
> See *Access Client*.

Access Client
> GFS kernel module on each node that communicates with the Meta Data Server and provides simultaneous access to a shared file system.

activating a configuration (RMS)
> Preparing an RMS configuration to be run on a cluster. This involves two major actions: first, the configuration is **generated** on the host where the configuration was created or edited; second, the configuration is **distributed** to all nodes affected by the configuration. The user can activate a configuration using PCS, the Wizard Tools, or the CLI.
>
> See also *generating a configuration (RMS),* distributing a configuration (RMS).

Administrative LAN
> In PRIMECLUSTER configurations, an Administrative LAN is a private local area network (LAN) on which machines such as the System Console and Cluster Console reside. Because normal users do not have access to the Administrative LAN, it provides an extra level of security. The use of an Administrative LAN is optional.
>
> See also public LAN.

API
> See Application Program Interface.

application (RMS)
> A resource categorized as a `userApplication` used to group resources into a logical collection.

Application Program Interface
> A shared boundary between a service provider and the application that uses that service.

application template (RMS)
> A predefined group of object definition value choices used by PCS, the Wizard Tools, or the PCS Wizard Kit to create object definitions for a specific type of application.

attribute (RMS)
> The part of an object definition that specifies how the base monitor acts and reacts for a particular object type during normal operations.

automatic switchover (RMS)
> The procedure by which RMS automatically switches control of a `userApplication` over to another node after specified conditions are detected.
>
> See also directed switchover (RMS), failover (RMS, SIS), switchover (RMS), symmetrical switchover (RMS).

availability
> Availability describes the need of most enterprises to operate applications via the Internet 24 hours a day, 7 days a week. The relationship of the actual to the planned usage time determines the availability of a system.

base cluster foundation (CF)
> This PRIMECLUSTER module resides on top of the basic OS and provides internal interfaces for the CF (Cluster Foundation) functions that the PRIMECLUSTER services use in the layer above.
>
> See also Cluster Foundation (CF).

base monitor (RMS)

The RMS module that maintains the availability of resources. The base monitor is supported by daemons and detectors. Each node being monitored has its own copy of the base monitor.

Cache Fusion

The improved interprocess communication interface in Oracle 9i that allows logical disk blocks (buffers) to be cached in the local memory of each node. Thus, instead of having to flush a block to disk when an update is required, the block can be copied to another node by passing a message on the interconnect, thereby removing the physical I/O overhead.

CCBR

See Cluster Configuration Backup and Restore.

CF

See Cluster Foundation (CF).

CF node name (CF)

The CF cluster node name, which is configured when a CF cluster is created.

child (RMS)

A resource defined in the configuration file that has at least one parent. A child can have multiple parents, and can either have children itself (making it also a parent) or no children (making it a leaf object).

See also resource (RMS), object (RMS), parent (RMS).

cluster

A set of computers that work together as a single computing source. Specifically, a cluster performs a distributed form of parallel computing.

See also RMS configuration (RMS).

Cluster Admin

A Java-based, OS-independent management tool for PRIMECLUSTER products such as CF, RMS, PCS, and SIS. Cluster Admin is available from the Web-Based Admin View interface.

See also Cluster Foundation (CF), Reliant Monitor Services (RMS), PRIMECLUSTER Configuration Services (PCS), Scalable Internet Services (SIS), Web-Based Admin View.

Cluster Configuration Backup and Restore
>   CCBR provides a simple method to save the current PRIMECLUSTER configuration information of a cluster node. It also provides a method to restore the configuration information.

Cluster Foundation (CF)
>   The set of PRIMECLUSTER modules that provides basic clustering communication services.
>
>   See also base cluster foundation (CF).

cluster interconnect (CF)
>   The set of private network connections used exclusively for PRIME-CLUSTER communications.

Cluster Join Services (CF)
>   This PRIMECLUSTER module handles the forming of a new cluster and the addition of nodes.

concatenated virtual disk (RCVM)
>   Concatenated virtual disks consist of two or more pieces on one or more disk drives. They correspond to the sum of their parts. Unlike simple virtual disks where the disk is subdivided into small pieces, the individual disks or partitions are combined to form a single large logical disk.
>
>   See also mirror virtual disk (RCVM), simple virtual disk (RCVM), striped virtual disk (RCVM), virtual disk.

Configuration Definition Language (PCS)
>   The syntax for PCS configuration templates.
>
>   See also PRIMECLUSTER Configuration Services (PCS).

configuration file (RMS)
>   The RMS configuration file that defines the monitored resources and establishes the interdependencies between them. The default name of this file is `config.us`.

console
>   See single console.

custom detector (RMS)
>   See detector (RMS).

custom type (RMS)
> See generic type (RMS).

daemon
> A continuous process that performs a specific function repeatedly.

database node (SIS)
> Nodes that maintain the configuration, dynamic data, and statistics in a SIS configuration.
>
> See also gateway node (SIS), service node (SIS), Scalable Internet Services (SIS).

detector (RMS)
> A process that monitors the state of a specific object type and reports a change in the resource state to the base monitor.

directed switchover (RMS)
> The RMS procedure by which an administrator switches control of a `userApplication` over to another node.
>
> See also automatic switchover (RMS), failover (RMS, SIS), switchover (RMS), symmetrical switchover (RMS).

distributing a configuration (RMS)
> The process of copying a configuration file and all of its associated scripts and detectors to all nodes affected by the configuration. This is normally done automatically when the configuration is **activated** using PCS, the Wizard Tools, or the CLI.
>
> See also activating a configuration (RMS), generating a configuration (RMS).

DOWN (CF)
> A node state that indicates that the node is unavailable (marked as down). A `LEFTCLUSTER` node must be marked as `DOWN` before it can rejoin a cluster.
>
> See also UP (CF), LEFTCLUSTER (CF), node state (CF).

ENS (CF)
> See Event Notification Services (CF).

environment variables
>Variables or parameters that are defined globally.

error detection (RMS)
>The process of detecting an error. For RMS, this includes initiating a log entry, sending a message to a log file, or making an appropriate recovery response.

Event Notification Services (CF)
>This PRIMECLUSTER module provides an atomic-broadcast facility for events.

failover (RMS, SIS)
>With SIS, this process switches a failed node to a backup node. With RMS, this process is known as switchover.
>
>See also automatic switchover (RMS), directed switchover (RMS), switchover (RMS), symmetrical switchover (RMS).

gateway node (SIS)
>Gateway nodes have an external network interface. All incoming packets are received by this node and forwarded to the selected service node, depending on the scheduling algorithm for the service.
>
>See also service node (SIS), database node (SIS), Scalable Internet Services (SIS).

GDS
>See Global Disk Services.

generating a configuration (RMS)
>The process of creating a single configuration file that can be distributed to all nodes affected by the configuration. This is normally done automatically when the configuration is **activated** using PCS, the RMS Wizards, or the CLI.
>
>See also activating a configuration (RMS), distributing a configuration (RMS).

GFS
>See Global File Services.

GLS
>See Global Link Services.

Global Disk Services

This optional product provides volume management that improves the availability and manageability of information stored on the disk unit of the Storage Area Network (SAN).

Global File Services

This optional product provides direct, simultaneous accessing of the file system on the shared storage unit from two or more nodes within a cluster.

Global Link Services

This PRIMECLUSTER optional module provides network high availability solutions by multiplying a network route.

generic type (RMS)

An object type which has generic properties. A generic type is used to customize RMS for monitoring resources that cannot be assigned to one of the supplied object types.

See also object type (RMS).

graph (RMS)

See system graph (RMS).

graphical user interface

A computer interface with windows, icons, toolbars, and pull-down menus that is designed to be simpler to use than the command-line interface.

GUI

See graphical user interface.

high availability

A system design philosophy in which redundant resources are employed to avoid single points of failure.

See also Reliant Monitor Services (RMS).

interconnect (CF)

See cluster interconnect (CF).

Internet Protocol address
A numeric address that can be assigned to computers or applications.

See also *IP aliasing*.

Internode Communications facility
This module is the network transport layer for all PRIMECLUSTER internode communications. It interfaces by means of OS-dependent code to the network I/O subsystem and guarantees delivery of messages queued for transmission to the destination node in the same sequential order unless the destination node fails.

IP address
See *Internet Protocol address*.

IP aliasing
This enables several IP addresses (aliases) to be allocated to one physical network interface. With IP aliasing, the user can continue communicating with the same IP address, even though the application is now running on another node.

See also *Internet Protocol address*.

JOIN (CF)
See *Cluster Join Services (CF)*.

keyword
A word that has special meaning in a programming language. For example, in the configuration file, the keyword `object` identifies the kind of definition that follows.

leaf object (RMS)
A bottom object in a system graph. In the configuration file, this object definition is at the beginning of the file. A leaf object does not have children.

LEFTCLUSTER (CF)
A node state that indicates that the node cannot communicate with other nodes in the cluster. That is, the node has left the cluster. The reason for the intermediate `LEFTCLUSTER` state is to avoid the network partition problem.

See also *UP (CF)*, *DOWN (CF)*, *network partition (CF)*, *node state (CF)*.

link (RMS)

Designates a child or parent relationship between specific resources.

local area network

See *public LAN*.

local node

The node from which a command or process is initiated.

See also *remote node, node*.

log file

The file that contains a record of significant system events or messages. The base monitor, wizards, and detectors can have their own log files.

MDS

See *Meta Data Server*.

message

A set of data transmitted from one software process to another process, device, or file.

message queue

A designated memory area which acts as a holding place for messages.

Meta Data Server

GFS daemon that centrally manages the control information of a file system (meta-data).

mirrored disks (RCVM)

A set of disks that contain the same data. If one disk fails, the remaining disks of the set are still available, preventing an interruption in data availability.

See also *mirrored pieces (RCVM)*.

mirrored pieces (RCVM)

Physical pieces that together comprise a mirrored virtual disk. These pieces include mirrored disks and data disks.

See also *mirrored disks (RCVM)*.

mirror virtual disk (RCVM)

Mirror virtual disks consist of two or more physical devices, and all output operations are performed simultaneously on all of the devices.

See also *concatenated virtual disk (RCVM)*, *simple virtual disk (RCVM)*, *striped virtual disk (RCVM)*, *virtual disk*.

mount point

The point in the directory tree where a file system is attached.

multihosting

Multiple controllers simultaneously accessing a set of disk drives.

native operating system

The part of an operating system that is always active and translates system calls into activities.

network partition (CF)

This condition exists when two or more nodes in a cluster cannot communicate over the interconnect; however, with applications still running, the nodes can continue to read and write to a shared device, compromising data integrity.

node

A host which is a member of a cluster. A computer node is the same as a computer.

node state (CF)

Every node in a cluster maintains a local state for every other node in that cluster. The node state of every node in the cluster must be either UP, DOWN, or LEFTCLUSTER.

See also *UP (CF)*, *DOWN (CF)*, *LEFTCLUSTER (CF)*.

object (RMS)

In the configuration file or a system graph, this is a representation of a physical or virtual resource.

See also *leaf object (RMS)*, *object definition (RMS)*, *object type (RMS)*.

object definition (RMS)

An entry in the configuration file that identifies a resource to be monitored by RMS. Attributes included in the definition specify properties of the corresponding resource. The keyword associated with an object definition is `object`.

See also *attribute (RMS)*, *object type (RMS)*.

object type (RMS)

A category of similar resources monitored as a group, such as disk drives. Each object type has specific properties, or attributes, which limit or define what monitoring or action can occur. When a resource is associated with a particular object type, attributes associated with that object type are applied to the resource.

See also *generic type (RMS)*.

online maintenance

The capability of adding, removing, replacing, or recovering devices without shutting or powering off the node.

operating system dependent (CF)

This module provides an interface between the native operating system and the abstract, OS-independent interface that all PRIMECLUSTER modules depend upon.

Oracle Real Application Clusters (RAC)

Oracle RAC allows access to all data in a database to users and applications in a clustered or MPP (massively parallel processing) platform. Formerly known as Oracle Parallel Server (OPS).

OSD (CF)

See *operating system dependent (CF)*.

parent (RMS)

An object in the configuration file or system graph that has at least one child.

See also *child (RMS), configuration file (RMS)*, *system graph (RMS)*.

PCS

See *PRIMECLUSTER Configuration Services (PCS)*.

PCS Wizard Kit (PCS)

RMS configuration products that have been designed for specific applications. Each component of the PCS Wizard Kit includes customized default settings, subapplications, detectors, and scripts. These application wizards also tailor the PCS interface to provide controls for the additional features.

See also *PCS*, *Reliant Monitor Services (RMS)*.

primary node (RMS)

The default node on which a user application comes online when RMS is started. This is always the node name of the first child listed in the `userApplication` object definition.

PRIMECLUSTER Configuration Services (PCS)

The graphical configuration interface for PRIMECLUSTER products. PCS uses standard templates written in Configuration Definition Language (CDL) to provide a user-friendly configuration environment for products such as RMS. The standard templates can be modified or replaced to provide a customized interface for specific applications or installations.

PRIMECLUSTER services (CF)

Service modules that provide services and internal interfaces for clustered applications.

private network addresses

Private network addresses are a reserved range of IP addresses specified by the Internet Assigned Numbers Authority. They may be used internally by any organization but, because different organizations can use the same addresses, they should never be made visible to the public internet.

private resource (RMS)

A resource accessible only by a single node and not accessible to other RMS nodes.

See also *resource (RMS)*, *shared resource*.

public LAN

The local area network (LAN) by which normal users access a machine.

See also *Administrative LAN*.

queue

See *message queue*.

redundancy

This is the capability of one object to assume the resource load of any other object in a cluster, and the capability of RAID hardware and/or RAID software to replicate data stored on secondary storage devices.

Reliant Monitor Services (RMS)

The package that maintains high availability of user-specified resources by providing monitoring and switchover capabilities.

remote node

A node that is accessed through a LAN or telecommunications line.

See also *local node, node*.

reporting message (RMS)

A message that a detector uses to report the state of a particular resource to the base monitor.

resource (RMS)

A hardware or software element (private or shared) that provides a function, such as a mirrored disk, mirrored disk pieces, or a database server. A local resource is monitored only by the local node.

See also *private resource (RMS), shared resource*.

resource definition (RMS)

See *object definition (RMS)*.

resource label (RMS)

The name of the resource as displayed in a system graph.

resource state (RMS)

Current state of a resource.

RMS

See *Reliant Monitor Services (RMS)*.

RMS commands (RMS)

Commands that enable RMS resources to be administered from the command line.

RMS configuration (RMS)

A configuration made up of two or more nodes connected to shared resources. Each node has its own copy of operating system and RMS software, as well as its own applications.

RMS Wizard Kit (RMS)

RMS configuration products that have been designed for specific applications. Each component of the Wizard Kit includes customized default settings, subapplications, detectors, and scripts. These application wizards also tailor the RMS Wizard Tools interface to provide controls for the additional features.

See also *RMS Wizard Tools (RMS)*, *Reliant Monitor Services (RMS)*.

RMS Wizard Tools (RMS)

A software package composed of various configuration and administration tools used to create and manage applications in an RMS configuration.

See also *RMS Wizard Kit (RMS)*, *Reliant Monitor Services (RMS)*.

SAN

See *Storage Area Network*.

Scalable Internet Services (SIS)

Scalable Internet Services is a TCP connection load balancer, and dynamically balances network access loads across cluster nodes while maintaining normal client/server sessions for each connection.

scalability

The ability of a computing system to dynamically handle any increase in work load. Scalability is especially important for Internet-based applications where growth caused by Internet usage presents a scalable challenge.

SCON

See *single console*.

script (RMS)

A shell program executed by the base monitor in response to a state transition in a resource. The script may cause the state of a resource to change.

service node (SIS)

Service nodes provide one or more TCP services (such as FTP, Telnet, and HTTP) and receive client requests forwarded by the gateway nodes.

See also *database node (SIS)*, *gateway node (SIS), Scalable Internet Services (SIS)*.

SF

See *Shutdown Facility*.

shared resource

A resource, such as a disk drive, that is accessible to more than one node.

See also *private resource (RMS)*, *resource (RMS)*.

Shutdown Facility

The Shutdown Facility provides the interface for managing the shutdown of cluster nodes when error conditions occur. The SF also cares for advising other PRIMECLUSTER products of the successful completion of node shutdown so that recovery operations can begin.

simple virtual disk (RCVM)

Simple virtual disks define either an area within a physical disk partition or an entire partition.

See also *concatenated virtual disk (RCVM)*, *mirror virtual disk (RCVM)*, *striped virtual disk (RCVM)*, *virtual disk*.

single console

The workstation that acts as the single point of administration for nodes being monitored by RMS. The single console software, SCON, is run from the single console.

SIS

See *Scalable Internet Services (SIS)*.

state

See *resource state (RMS)*.

Storage Area Network

The high-speed network that connects multiple, external storage units and storage units with multiple computers. The connections are generally fiber channels.

striped virtual disk (RCVM)

Striped virtual disks consist of two or more pieces. These can be physical partitions or further virtual disks (typically a mirror disk). Sequential I/O operations on the virtual disk can be converted to I/O operations on two or more physical disks. This corresponds to RAID Level 0 (RAID0).

See also *concatenated virtual disk (RCVM)*, *mirror virtual disk (RCVM)*, *simple virtual disk (RCVM)*, *virtual disk*.

switchover (RMS)

The process by which RMS switches control of a userApplication over from one monitored node to another.

See also *automatic switchover (RMS)*, *directed switchover (RMS)*, *failover (RMS, SIS)*, *symmetrical switchover (RMS)*.

symmetrical switchover (RMS)

This means that every RMS node is able to take on resources from any other RMS node.

See also *automatic switchover (RMS)*, *directed switchover (RMS)*, *failover (RMS, SIS)*, *switchover (RMS)*.

system graph (RMS)

A visual representation (a map) of monitored resources used to develop or interpret the configuration file.

See also *configuration file (RMS)*.

template

See *application template (RMS)*.

type

See *object type (RMS)*.

UP (CF)

A node state that indicates that the node can communicate with other nodes in the cluster.

See also *DOWN (CF)*, *LEFTCLUSTER (CF)*, *node state (CF)*.

virtual disk

> With virtual disks, a pseudo device driver is inserted between the highest level of the OS logical Input/Output (I/O) system and the physical device driver. This pseudo device driver then maps all logical I/O requests on physical disks.
>
> See also *concatenated virtual disk (RCVM)*, *mirror virtual disk (RCVM)*, *simple virtual disk (RCVM)*, *virtual disk*.

Web-Based Admin View

> A Java-based, OS-independent interface to PRIMECLUSTER management components.
>
> See also *Cluster Admin*.

wizard (RMS)

> An interactive software tool that creates a specific type of application using pretested object definitions. An enabler is a type of wizard.

Wizard Kit (RMS)

> See *PCS Wizard Kit (PCS)*, *RMS Wizard Kit (RMS)*.

Wizard Tools (RMS)

> See *RMS Wizard Tools (RMS)*.

Abbreviations

AC
> Access Client

API
> application program interface

AS
> Adaptive Services

bm
> base monitor

CCBR
> Cluster Configuration Backup/Restore

CDL
> Configuration Definition Language

CF
> Cluster Foundation or Cluster Framework

CIM
> Cluster Integrity Monitor

CIP
> Cluster Interconnect Protocol

CLI
> command-line interface

CLM
> Cluster Manager

CRM
> Cluster Resource Management

DLPI
> Data Link Provider Interface

Abbreviations

ENS
Event Notification Services

GDS
Global Disk Services

GFS
Global File Services

GLS
Global Link Services

GUI
graphical user interface

HA
high availability

ICF
Internode Communication Facility

I/O
input/output

JOIN
cluster join services module

LAN
local area network

MDS
Meta Data Server

MIB
Management Information Base

MIPC
Mesh Interprocessor Communication

NIC
network interface card

NSM

Node State Monitor

OSD

operating system dependent

PAS

Parallel Application Services

PCS

PRIMECLUSTER Configuration Services

RCCU

Remote Console Control Unit

RCFS

PRIMECLUSTER File Share

RCI

Remote Cabinet Interface

RCVM

PRIMECLUSTER Volume Manager

RMS

Reliant Monitor Services

SA

Shutdown Agent

SAN

Storage Area Network

SCON

single console software

SD

Shutdown Daemon

SF

Shutdown Facility

SIS
Scalable Internet Services

VIP
Virtual Interface Provider

Figures

Tables

Index

Comments
Suggestions
Corrections

Submitted by

Comments on PRIMECLUSTER™
         Reliant Monitor Services (RMS) (Solaris®, Linux®)
         Troubleshooting Guide

Comments
Suggestions
Corrections

Submitted by

Comments on PRIMECLUSTER™
      Reliant Monitor Services (RMS) (Solaris®, Linux®)
      Troubleshooting Guide