

# PRIMECLUSTER™

Reliant Monitor Services (RMS)  
Configuration and Administration Guide (Solaris)

## **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](http://www.cognitas.de)

## **Copyright and Trademarks**

Copyright © 2002, 2003, Fujitsu Siemens Computers Inc. and Fujitsu LIMITED.

All rights reserved.

Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

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

---

Preface

---

Introduction

---

Configuration

---

Administration

---

Troubleshooting

---

Appendix—Object types

---

Appendix—Attributes

---

Appendix—Environment variables

---

Manual pages

---

Glossary

**Continued** ►



---

Abbreviations

---

Figures

---

Tables

---

Index

---



---

# Contents

<b>1</b>	<b>Preface</b>	<b>1</b>
1.1	About this manual	1
1.2	Documentation	2
1.2.1	Suggested documentation	3
1.3	Conventions	3
1.3.1	Notation	3
1.3.1.1	Prompts	4
1.3.1.2	Manual page section numbers	4
1.3.1.3	The keyboard	4
1.3.1.4	Typefaces	4
1.3.1.5	Example 1	5
1.3.1.6	Example 2	5
1.3.2	Command syntax	5
1.4	Important	6
<b>2</b>	<b>Introduction</b>	<b>7</b>
2.1	PRIMECLUSTER overview	7
2.2	How RMS provides high availability	8
2.3	How RMS Wizards provide easy configuration	10
2.3.1	RMS Wizard Tools	13
2.3.2	RMS Application Wizards	13
2.4	Cluster Admin	14
2.5	RMS components	14
2.5.1	Base monitor	14
2.5.2	Detectors and states	15
2.5.3	Scripts	16
2.5.4	RMS CLI	17
2.6	Object types	19
2.7	Attributes	20
2.8	Environment variables	20
2.9	Directory structure	22
<b>3</b>	<b>Configuration</b>	<b>25</b>
3.1	Overview	25
3.2	RMS Wizards configuration	26
3.2.1	RMS Wizards types	26
3.2.1.1	Turnkey wizards	27
3.2.1.2	Resource wizards	27
3.2.2	Site preparation	28
3.2.2.1	NFS Lock Failover	28
3.2.2.2	Important files	29

## Contents

---

3.2.3	Creating and editing a configuration . . . . .	30
3.2.4	Using the wizard menus . . . . .	31
3.2.4.1	Main configuration menu . . . . .	31
3.2.4.2	Main configuration menu when RMS is inactive . . . . .	32
3.2.4.3	Main configuration menu when RMS is running . . . . .	36
3.2.4.4	Secondary menus . . . . .	36
3.2.5	Basic and non-basic settings . . . . .	37
3.2.6	Usability . . . . .	39
3.2.7	Configuration elements . . . . .	40
3.2.7.1	Scripts . . . . .	40
3.2.7.2	Detectors . . . . .	41
3.2.7.3	RMS objects . . . . .	41
3.3	Sample configuration . . . . .	42
3.3.1	Creating an application . . . . .	42
3.3.2	Activating a configuration . . . . .	57
3.3.3	Creating a second application . . . . .	59
3.3.4	Activating the configuration a second time . . . . .	72
3.4	Further reading . . . . .	75
<b>4</b>	<b>Administration . . . . .</b>	<b>77</b>
4.1	Overview . . . . .	77
4.2	Using Cluster Admin . . . . .	77
4.2.1	Starting Cluster Admin . . . . .	77
4.2.2	Logging in . . . . .	79
4.2.3	Main screen . . . . .	81
4.2.4	RMS main window . . . . .	83
4.2.4.1	RMS tree . . . . .	83
4.2.4.2	Configuration information or object attributes . . . . .	85
4.2.4.3	Command pop-ups . . . . .	86
4.2.4.4	Switchlogs and application logs . . . . .	88
4.2.5	RMS graphs . . . . .	93
4.2.5.1	RMS full graph . . . . .	93
4.2.5.2	Application graph . . . . .	95
4.2.5.3	Sub-application graph . . . . .	96
4.2.5.4	Composite sub-application graph . . . . .	97
4.2.5.5	Configuration information and the graphs . . . . .	98
4.2.5.6	Command pop-ups . . . . .	99
4.2.5.7	RMS graph customization . . . . .	100
4.2.5.8	RMS GUI after it is shut down . . . . .	103
4.2.6	RMS clusterwide table . . . . .	104
4.2.6.1	Command pop-ups . . . . .	107
4.2.7	Changing the RMS configuration . . . . .	108
4.3	RMS procedures . . . . .	110
4.3.1	Starting RMS . . . . .	111



4.3.2	Stopping RMS . . . . .	115
4.3.3	Starting an application . . . . .	119
4.3.4	Switching an application . . . . .	121
4.3.5	Activating an application . . . . .	123
4.3.6	Taking an application offline . . . . .	124
4.3.7	Clearing a fault . . . . .	125
4.3.8	Clearing a SysNode Wait state . . . . .	126
4.3.9	Displaying environment variables . . . . .	127
4.3.10	Displaying application states . . . . .	130
4.3.11	Viewing switchlogs . . . . .	131
4.3.12	Viewing application logs . . . . .	131
4.3.13	Viewing GUI messages . . . . .	131
<b>5</b>	<b>Troubleshooting . . . . .</b>	<b>133</b>
5.1	Overview . . . . .	134
5.2	Debug and error messages . . . . .	135
5.3	Log files . . . . .	137
5.4	Using the log viewer . . . . .	139
5.4.1	Search based on resource . . . . .	142
5.4.2	Search based on time . . . . .	143
5.4.3	Search based on keyword . . . . .	144
5.4.4	Search based on severity levels . . . . .	145
5.5	Using hvdump command . . . . .	146
5.6	Specifying the log level . . . . .	147
5.7	Interpreting log files . . . . .	149
5.8	System log . . . . .	150
5.9	RMS Wizard logs . . . . .	151
5.9.1	RMS Wizards detector logging . . . . .	153
5.9.2	Modifying levels while RMS is running . . . . .	154
5.10	RMS error code description . . . . .	155
5.11	Errors in the switchlog . . . . .	156
5.12	Fatal error messages . . . . .	241
5.13	Additional error messages . . . . .	252
5.14	RMS troubleshooting . . . . .	274
<b>6</b>	<b>Appendix—Object types . . . . .</b>	<b>277</b>
<b>7</b>	<b>Appendix—Attributes . . . . .</b>	<b>279</b>
<b>8</b>	<b>Appendix—Environment variables . . . . .</b>	<b>295</b>
8.1	Global environment variables . . . . .	295
8.2	Local environment variables . . . . .	298

# Contents

---

<b>9</b>	<b>Manual pages</b>	<b>301</b>
9.1	CCBR	301
9.2	CF	301
9.3	CFS	302
9.4	CIP	302
9.5	Monitoring Agent	303
9.6	PAS	303
9.7	RCVM	303
9.8	Resource Database	304
9.9	RMS	305
9.10	RMS Wizards	306
9.11	SCON	306
9.12	SF	306
9.13	SIS	307
9.14	Web-Based Admin View	307
<b>Glossary</b>		<b>309</b>
<b>Abbreviations</b>		<b>325</b>
<b>Figures</b>		<b>329</b>
<b>Tables</b>		<b>335</b>
<b>Index</b>		<b>337</b>

---

# 1 Preface

This manual describes how Reliant<sup>®</sup> Monitor Services (RMS) functions, configuring RMS using the RMS Wizards, and administering RMS by means of the Cluster Admin GUI. RMS is a software monitor designed to guarantee the high availability of applications in a cluster of nodes.

This manual provides an introduction to the product and describes the configuration and administration tasks that are essential if RMS is to operate properly.

The manual is aimed at system administrators and programmers familiar with installing and maintaining RMS configurations.

Persons who configure and administer RMS should be familiar with the following system functions and components:

- PRIMECLUSTER family of products
- Solaris operating system
- Non PRIMECLUSTER products

## 1.1 About this manual

This manual is structured as follows:

- The Chapter “Introduction” contains general information on RMS and introduces the PRIMECLUSTER family of products.
- The Chapter “Configuration” describes how to configure RMS using the RMS Wizards.
- The Chapter “Administration” discusses how to administer RMS by means of the Cluster Admin GUI.
- The Chapter “Troubleshooting” describes how to troubleshoot RMS and lists error messages, their causes, and resolutions.
- The Chapter “Appendix—Object types” lists all the object types that are supplied with RMS.
- The Chapter “Appendix—Attributes” lists the attributes that are required for each object type.

- The Chapter “Appendix—Environment variables” describes the RMS environment variables.
- The Chapter “Manual pages” lists the manual pages for PRIMECLUSTER.

## 1.2 Documentation

The documentation listed in this section contains information relevant to PRIMECLUSTER and can be ordered through your sales representative.

In addition to this manual, the following manuals are also available for PRIMECLUSTER:

- *Installation Guide (Solaris)*—Provides instructions for installing PRIMECLUSTER.
- *Cluster Foundation (CF) Configuration and Administration Guide (Solaris)*—Provides instructions for configuring and administering the PRIMECLUSTER Cluster Foundation.
- *Concepts Guide (Solaris, Linux)*—Provides conceptual details on the PRIMECLUSTER family of products.
- *Scalable Internet Services (SIS) Configuration and Administration Guide (Solaris, Linux)*—Provides information on configuring and administering Scalable Internet Services.
- *Global Disk Services Configuration and Administration Guide (Solaris)*—Provides information on configuring and administering Global Disk Services (GDS).
- *Global File Services Configuration and Administration Guide (Solaris)*—Provides information on configuring and administering Global File Services (GFS).
- *Global Link Services Configuration and Administration Guide: Redundant Line Control Function (Solaris)*—Provides information on configuring and administering the redundant line control function for Global Link Services (GLS).
- *Global Link Services Configuration and Administration Guide: Multipath Function (Solaris)*—Provides information on configuring and administering the multipath function for Global Link Services (GLS).
- *Web-Based Admin View Operation Guide (Solaris)*—Provides information on using the Web-Based Admin View management GUI.

- *SNMP Reference Manual (Solaris)*—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 topics like the configuration of file systems and IP addresses, or the different kinds of wizards.

### 1.2.1 Suggested documentation

The following manuals contain information relevant to RMS administration and can be ordered through your sales representative (not available in all areas):

- *ANSI C Programmer's Guide*
- *LAN Console Installation, Operation and Maintenance*
- *Terminal TM100/TM10 Operating Manual*
- *PRIMEPOWER User's Manual* (operating manual)



Your sales representative will need your operating system release and product version to place your order.

## 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 *node#* 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.
<i>Italic</i>	Variables that you must replace with an actual value.
<b>Bo1d</b>	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

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



Contains important information about the subject at hand.



### **Caution**

Indicates a situation that can cause harm to data.



---

## 2 Introduction

This chapter contains general information on Reliant Monitor Services (RMS), introduces the PRIMECLUSTER family of products, details how RMS, RMS Application Wizards, and RMS Wizard Tools products work together to produce high-availability configurations, and introduces Cluster Admin.

This chapter discusses the following:

- The Section “PRIMECLUSTER overview” describes how RMS functions in relationship to the PRIMECLUSTER family of products.
- The Section “How RMS provides high availability” describes how RMS supplies high availability.
- The Section “How RMS Wizards provide easy configuration” details the RMS Wizard products: RMS Wizard Tools and RMS Application Wizards.
- The Section “Cluster Admin” introduces the Cluster Admin graphical user interface (GUI).
- The Section “Object types” introduces the RMS object types.
- The Section “Attributes” defines the RMS attributes.
- The Section “Environment variables” lists the RMS environment variables.
- The Section “Directory structure” lists and describes the RMS directory structure.

### 2.1 PRIMECLUSTER overview

The PRIMECLUSTER family of products is an integrated set of cluster services, including high availability, scalability, parallel application support, cluster file system, cluster volume management and administration. Figure 1 shows the types of services found in PRIMECLUSTER.

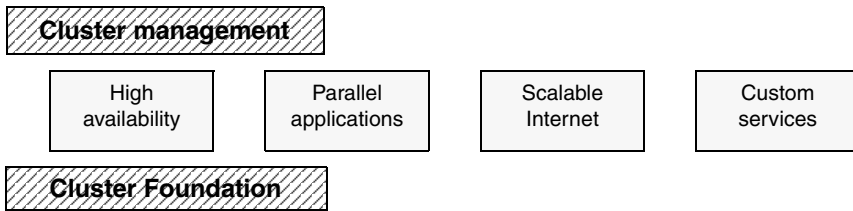


Figure 1: Overview of PRIMECLUSTER

The sections that follow focus on the role of the following PRIMECLUSTER products as they relate to RMS:

- **RMS**—This high availability manager is a software monitor that provides high availability (HA) for customer applications in a cluster of nodes. Its task is to monitor systems and application resources, to identify any failures, and to provide application availability virtually without interruption in the event of any such failures.
- **RMS Wizard Tools and RMS Application Wizards**—These configuration products are used to create RMS configurations capable of controlling any number of user applications.
- **Cluster Admin**—The Cluster Admin GUI is the primary administrative tool for RMS.

## 2.2 How RMS provides high availability

RMS provides high availability of a customer's application by controlling and monitoring the state of all resources in use by a given application. Within RMS, each resource used by an application is represented as an object, and each object is configured with the following:

- Scripts
- Detectors
- Dependent resources

RMS achieves control of individual resources by running scripts that are configured within RMS for each resource. One set of scripts brings the resource *Online*, while another brings the resource *Offline*. RMS monitors each resource by using detector processes that report resource states to the RMS base monitor process.

The set of resources in use by an application are known collectively as a `userApplication` object. The nodes in a cluster that may run an application are known as `SysNode` objects.

Bringing a `userApplication` object to an `Online` state, along with all of its dependent resources, is called online processing. Bringing a `userApplication` object to an `Offline` state, along with all of its dependent resources, is called offline processing.

Normally a `userApplication` object is only allowed to be `Online` on one `SysNode` at a time. Exceptions to this rule are shared objects like OPS vdisk. If a fault occurs within a resource used by a `userApplication` object, then the `userApplication` object is switched over to another node in the cluster. A switchover of a `userApplication` object involves running offline processing on the node where the application is currently `Online` and then running online processing on a new node. RMS also has the ability to recover a resource locally; that is, bring a faulted resource back to an `Online` state without switching the entire `userApplication` to another cluster node.

In addition to monitoring the state of each resource, RMS also monitors the state of each node in the cluster. If RMS loses contact with a node, or if the node represented by the `SysNode` object crashes, then all `userApplication` objects that were online on the crashed node are switched over to a surviving node in the cluster.

RMS works together with the PRIMECLUSTER Shutdown Facility to ensure that cluster nodes that are no longer responding are truly down before bringing the `userApplication` object to an `Online` state on another node. This helps to ensure data consistency.

An RMS instance runs on each node in the cluster. Each of these instances communicate with one another to coordinate the actions associated with each configured `userApplication`. RMS must be running on a given node for application failover to work.

## Overview of configuration

In addition to being able to configure and fine-tune the application, those performing RMS configurations must also know how RMS works and what RMS needs in order to function properly.

To properly configure RMS to control an application, you must do the following:

- Define the set of resources used by the application, including:
  - File systems
  - File systems of Global File Systems (GFS)
  - Running processes
  - IP addresses
  - IP addresses of Global Link Services (GLS)
- Define the nodes that a `userApplication` object will be available to run on, including the priority order for application switchover.
- Define the relationship between each resource and its dependent resources; in other words, which file system depends on which virtual or physical disk, which processes depend on which file systems, and so forth.
- Define the relationship between the applications being controlled; for example, which applications are started before others are allowed to start.
- Provide scripts to bring each resource `Online` and `Offline`.
- Provide a detector to determine the state of each resource.

Configuring the above set of requirements by hand can be quite time consuming and prone to errors. This is why the RMS Wizards were developed.

## 2.3 How RMS Wizards provide easy configuration

PRIMECLUSTER provides the RMS Wizards to allow the creation of flexible and quality-tested RMS configurations. The RMS Wizards present a simple user interface that prompts you for details regarding the applications. The RMS Wizards are designed to easily configure RMS for certain popular applications such as Oracle or SAP R/3, and they are flexible enough to create full RMS configurations that can control any other type of application.

Specialists skilled in popular applications and in RMS worked together to create the RMS Wizards. The RMS Wizards are broken up into the following separate products:

- RMS Wizard Tools

- RMS Application Wizards

Figure 2 depicts the relationship between RMS, RMS Wizard Tools, and RMS Application Wizards.

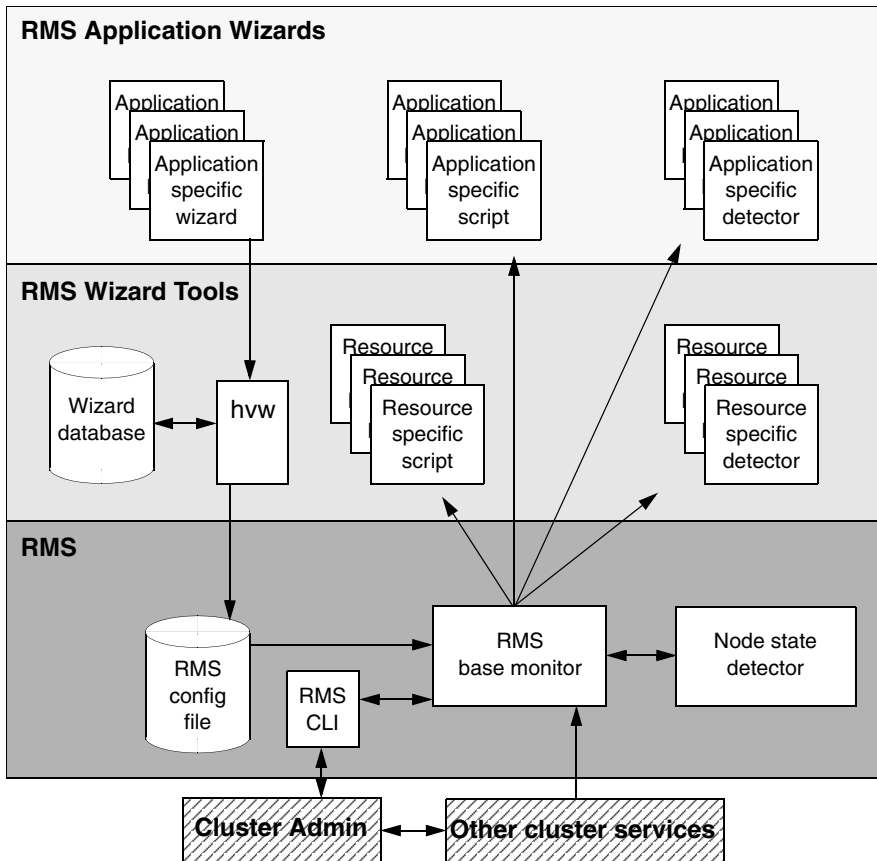


Figure 2: Relationship between RMS and RMS Wizards

### 2.3.1 RMS Wizard Tools

The RMS Wizard Tools provides the following for basic resource types (such as file systems and IP addresses):

- Online scripts
- Offline scripts
- Detectors

In addition to the basic resource support, the RMS Wizard Tools package contains the `hvw` command which is the entry point to the user configuration interface. The `hvw` interface provides a simple menu-driven interface to allow a user to enter information specific to applications placed under the control of RMS. `hvw` also provides an interface through which application-specific knowledge can be dynamically added to provide turnkey solutions for those applications typically found in the data center. These application-specific modules are provided by the RMS Application Wizards products.

### 2.3.2 RMS Application Wizards

The RMS Application Wizards provide application knowledge modules which can be used by the `hvw` command. The knowledge modules provide `hvw` with information specific to popular applications, which greatly eases the configuration task. The following are also provided for specific applications:

- Online scripts
- Offline scripts
- Detectors



For information on the availability of the RMS Application Wizards, contact your local customer support service or refer to the RMS Wizards documentation package (refer to Section “Further reading” for more details).

## 2.4 Cluster Admin

The Cluster Admin GUI is the primary administrative tool for RMS. For RMS, it allows users full access to the application control functions of RMS, including the following:

- Application startup
- Application shutdown
- Manual application switchover
- Visual cues for resource and application fault isolation
- Fault clearing capability
- RMS startup
- RMS shutdown

## 2.5 RMS components

The RMS product is made up of the following software components that run on each node in the cluster:

- Base monitor
- Detectors
- Scripts
- RMS CLI

### 2.5.1 Base monitor

The base monitor process is the decision-making segment of the RMS process group. It has the following functions:

- Stores the current configuration of resources as depicted by objects, their attributes, and their interdependent relationships
- Receives requests from the RMS command line interface (CLI) to take actions
- Receives input from detectors that report state changes



- Launches scripts to bring applications and their dependent resources `Online` or `Offline`
- Dictates the sequencing of the resource state changes to ensure resources and applications are brought `Online` or `Offline` in the correct order
- Initiates and controls automatic application switchover as required by a CLI request or in case of a resource or node failure
- Performs various administrative functions

## 2.5.2 Detectors and states

Detectors are independent processes that monitor specific sets of resources in order to determine their state. The detector does not determine if the current state of a resource is the correct state or not (for example, if a resource is `Offline` but is supposed to be `Online`), that is the role of the base monitor.

Detectors can report the following states to the base monitor:

<code>Online</code>	Resource is enabled and ready for use.
<code>Offline</code>	Resource is disabled and should not be used.
<code>Faulted</code>	Resource encountered an error condition.
<code>Warning</code>	Resource has exceeded some warning threshold.
<code>Standby</code>	Resource is in such a state that it can be quickly brought <code>Online</code> when needed.
<code>OfflineFaulted</code>	Resource is <code>OfflineFault</code> , signifying a fault occurred in the past and has not yet been cleared.

### 2.5.3 Scripts

RMS uses scripts to perform actions such as moving a resource from one state to another (for example, from `Offline` to `Online`). The two types of scripts are as follows:

- Request-triggered scripts initiate a state change to a resource.

The request-triggered scripts are as follows:

- `InitScript` —Runs only once when RMS is first started
- `PreCheckScript`—Determines if `Online` processing is needed or possible
- `PreOfflineScript`—Prepares a transition to an `Offline` state
- `OfflineScript`—Transitions a resource to an `Offline` state
- `PreOnlineScript`—Prepares a transition to an `Online` state
- `OnlineScript`—Transitions a resource to an `Online` state

- State-triggered scripts react to specific events.

The state-triggered scripts are as follows:

- `PostOnlineScript`—Reaction to the transition to the `Online` state
- `PostOfflineScript`—Reaction to the transition to the `Offline` state
- `OfflineDoneScript`—Reaction to a `userApplication` reaching the `Offline` state
- `FaultScript`—Reaction to a resource transitioning to the `Faulted` state
- `WarningScript`—Reaction to a detector reporting the `Warning` state

## 2.5.4 RMS CLI

The primary interface for configuring RMS is the RMS Wizards and the primary interface for administering RMS is the Cluster Admin GUI. Both the RMS Wizards and the Cluster Admin call the RMS CLI, and, under certain conditions, you may find the CLI useful. For example, you might want to manually switch a user application to another node in the cluster. You can do this by using the following CLI command:

► # `hvs witch userApplication SysNode`

In this case, *userApplication* is the user application that the user wants to switch to the system node *SysNode*.

Table 1 lists the RMS CLI commands available to administrators. Refer to the Chapter “Manual pages” for additional information on RMS CLI commands.

Command	Function
<code>hvassert</code>	The <code>hvassert</code> command tests an RMS resource for a specified resource state. It can be used in scripts when a resource must achieve a specified state before the script can issue the next command.
<code>hvcm</code>	The <code>hvcm</code> command starts the base monitor and the detectors for all monitored resources. In most cases, it is not necessary to specify options to the <code>hvcm</code> command. The base monitor is the decision-making module of RMS. It controls the configuration and access to all RMS resources. If a resource fails, the base monitor analyzes the failure and initiates the appropriate action according to the specifications for the resource in the configuration file.
<code>hvconfig</code>	The <code>hvconfig</code> command performs two tasks: displaying the current RMS configuration or sending the current configuration to an output file. The output of the <code>hvconfig</code> command is equivalent to the running RMS configuration file, but does not include any comments that are in the original file. Also, the order in which the resources are listed in the output might vary from the actual configuration file.
<code>hvdisp</code>	The <code>hvdisp</code> command displays information about the current configuration for RMS resources.

Table 1: Available CLI commands

Command	Function
hvdist	The <code>hvdist</code> command distributes the configuration file to all nodes within an RMS configuration.
hvdump	The <code>hvdump</code> command gets debugging information about RMS on the local node.
hvgdmake	The <code>hvgdmake</code> command makes (compiles) a custom detector so that it can be used in the RMS configuration. The user first prepares a source file for the detector, which must be a file with a <code>.c</code> extension.
hvlogclean	The <code>hvlogclean</code> command saves old log files into a subdirectory whose name is the time RMS was last started (unless the <code>-d</code> option is used to delete the old log files instead). Regardless, <code>hvlogclean</code> creates a clean set of log files even while RMS is running.
hvrclv	The <code>hvrclv</code> command sets the RMS default-start run level to 3 to allow for the system processes started in the remote-file-sharing state as well as any user application resources started in run level 3. The <code>hvrclv</code> command can be used to reset the RMS default start run level back to the original run level 2. The <code>hvrclv</code> command is typically called from <code>pkgadd</code> to automatically adjust the RMS start run level for those customers who have a default system run level of 3.
hvshut	The <code>hvshut</code> command shuts down RMS on one or more nodes in the configuration. The base monitor on the local node sends a message to other online nodes indicating which node or nodes will be shut down.
hvswitch	The <code>hvswitch</code> command manually switches control of a user application resource from one system node to another in the RMS configuration. The resource being switched must be of type <code>userApplication</code> . The system node must be of type <code>SysNode</code> .

Table 1: Available CLI commands

Command	Function
hvthrottle	The <code>hvthrottle</code> command prevents multiple scripts within a configuration file from running at the same time.
hvutil	The <code>hvutil</code> command is the general administration interface to RMS. It performs various resource administration tasks, such as dynamically setting logging levels, bringing a resource <code>Offline</code> , clearing faulted resources or hung cluster nodes in the <code>Wait</code> state, and setting detector time periods, and so forth.

Table 1: Available CLI commands

## 2.6 Object types

An object type represents a group of similar resources that are monitored by the same detector (for example, all disk drives). Using the RMS Wizards, you can create configuration files, which contain objects of various types that represent resources or groups of resources that are monitored by RMS.

The supported types are as follows:

- `SysNode`
- `userApplication`
- `gResource`
- `andOp`
- `orOp`
- `controller`

Refer to Chapter “Appendix—Object types” for the supported types, their required attributes, and a description of each object.



This information is provided for reference material. These objects are created by the RMS Wizards during the *Configuration—Generate* phase of the configuration process (refer to the Chapter “Configuration”).

## 2.7 Attributes

An attribute is the part of an object definition that specifies how the base monitor acts and reacts for a particular resource during normal operation. An attribute can include a device name and configuration scripts. Users can specify attributes in any order in the object definition.

Refer to Chapter “Appendix—Attributes” for the supported types, their associated values, and a description of each attribute.



This information is provided for reference material. The values are determined by the RMS Wizards during the *Configuration—Generate* phase of the configuration process (refer to the Chapter “Configuration”).

## 2.8 Environment variables

The `ENV` and `ENVL` objects contain the environment variables used by RMS. The values for the environment variables are found in the `hvenv` and `hvenv.local` files. The default values of the environment variables are found in `hvenv`. They can be redefined in the `hvenv.local` command file. The `hvenv` and `hvenv.local` files are located in the `RELIANT_PATH/bin` directory.

The two types of environment variables are as follows:

- Global variables (can be seen using the command `hvdisp ENV`). Global variables must have the same setting on all nodes in the cluster.
- Local variables (can be seen using the command `hvdisp ENVL`). Local variables can differ from node to node.



Global variable settings (`ENV`) are included into the configurations checksum which is verified among all nodes of the cluster during startup of the base monitor.

You cannot define the `ENV` or `ENVL` objects in the configuration file. These are automatically created and loaded with values when the RMS base monitor starts up.

The global environment variables (`ENV`) are as follows:

- `HV_AUTOSTARTUP_IGNORE`
- `HV_AUTOSTARTUP_WAIT`
- `HV_CHECKSUM_INTERVAL`

- HV\_LOG\_ACTION\_THRESHOLD
- HV\_LOG\_WARN\_THRESHOLD
- HV\_RCSTART
- RELIANT\_LOG\_LIFE
- RELIANT\_LOG\_PATH
- RELIANT\_PATH
- RELIANT\_SHUT\_MIN\_WAIT

The local environment variables (ENVL) are as follows:

- HV\_CONNECT\_TIMEOUT
- HV\_LOG\_ACTION
- HV\_MAX\_HVDISP\_FILE\_SIZE
- HV\_MAXPROC
- HV\_SYSLOG\_USE
- RELIANT\_HOSTNAME
- RELIANT\_INITSCRIPT
- RELIANT\_STARTUP\_PATH
- SCRIPTS\_TIME\_OUT

Refer to the Chapter “Appendix—Environment variables” for a description of all global and local environment variables.

### Setting environment variables

When RMS starts, it reads the values of environment variables from `hvenv` and `hvenv.local` and initializes the ENV and ENVL nodes respectively. To set the values of environment variables before starting RMS, the variables have to be specified in the `hvenv` and `hvenv.local` files.



A `/tmp` directory that is nearly full may result in RMS errors because `hvenv` uses this directory to sort RMS environment variables.

You can change the `hvenv.local` file on a node in the cluster, but the `hvenv` file must not be changed on any node. To activate your changes, you must stop RMS and restart it.

**Caution**

RMS environment variables cannot be set in the user environment explicitly. Doing so can cause RMS to lose environment variables settings.

The values of environment variables are specified as export directives in these files. An example of an export directive would be as follows:

```
export SCRIPTS_TIME_OUT=200
```

You should change environment variables before running the configuration file. You can view the values for environment variables by using the following commands while RMS is running:

- `hvdisp ENV`
- `hvdisp ENVL`

## 2.9 Directory structure

RMS software consists of a number of executables, scripts, files, and commands. Table 2 illustrates the directory structure of the RMS software after it has been correctly installed.

Name	Contents
RELIANT_PATH	The base directory default is: /opt/SMAW/SMAWRrms/
RELIANT_PATH/bin	Executables. These include detectors, commands, and scripts.
RELIANT_PATH/build	Use this as a work area for configuration files.
RELIANT_PATH/etc	Files that control the RMS environment.
RELIANT_PATH/include	RMS include files (header files) that are used by detectors and configuration files.

Table 2: RMS base directory structure



<b>Name</b>	<b>Contents</b>
RELIANT_PATH/lib	RMS runtime libraries.
RELIANT_PATH/tab	Tables used to restart detectors.
RELIANT_PATH/us	RMS source files. The names of the files in this directory are reserved and should not be used to name any configuration files that the user may create.

Table 2: RMS base directory structure

Table 3 illustrates the directory structure for the RMS log files.

Name	Contents
RELIANT_LOG_PATH	The log directory default is /var/opt/SMAWRrms/log. This contains files that can be used for RMS analyzing and debugging. Detectors and userApplications create log files here when they are started.

Table 3: Log directory structure

---

## 3 Configuration

This chapter describes how to configure high availability for customer applications using the RMS Wizards.

This chapter discusses the following:

- The Section “Overview” gives a brief overall description of the configuration procedure.
- The Section “RMS Wizards configuration” deals with some general aspects of setting up high-availability configurations with the aid of the tools known as the RMS Wizards.
- The Section “Sample configuration” gives an example that shows how a configuration is set up using the RMS Wizards to achieve high availability for two simple applications.
- The Section “Further reading” provides some information on documents for further reading about the RMS Wizards.

### 3.1 Overview

The Chapter “Introduction” describes the components necessary for configuring applications for high availability. It is extremely important that you define applications and the resources that are used by them. Resources are entities like disks, file systems, processes, IP addresses, and so forth.

This definition also needs to include the following information:

- How the applications and their resources are related to each other
- What scripts bring resources `Online` and `Offline`
- Which detectors monitor the state of which resources

If the node should fail to be available, the node that is to take its place must have been defined beforehand so that the applications depending on this node are able to continue operating with minimal interruption. Once the necessary information is defined, you can then set up an RMS configuration. A configuration of this magnitude, however, requires a great deal of expert knowledge to manually configure. This is why we developed the RMS Wizards—to greatly simplify the configuration process.

## 3.2 RMS Wizards configuration

The RMS Wizards are tools that allow you to set up an RMS configuration in a way that is simple, flexible, and quality-tested. Furthermore, these tools conform to a well-documented, standard design.

To configure RMS with the RMS Wizards, you supply information concerning the applications by means of a user interface consisting of a set of menus displayed on the system screen. The RMS Wizards use this information to set up a complete RMS configuration.

The following sections describe these wizards and the way they are used to configure high availability from a general point of view. This is followed by an example that shows how a wizard-based configuration is carried out.

### 3.2.1 RMS Wizards types

The RMS Wizards are divided into two categories. The two categories are necessary because configuring for high availability requires that you deal with applications and their resources.

The two basic types of wizards are as follows:

- **RMS Wizard Tools**—These resource-oriented wizards were developed to provide scripts and detectors for basic resources like file systems or IP addresses.
- **RMS Application Wizards**—These application-oriented wizards were developed to cover complete applications. Wizards of this type are, for instance, the R/3 wizard or the ORACLE wizard. These wizards perform their tasks on the basis of the turnkey concept.

The wizard naming convention provides for both turnkey-wizard style and resource-wizard style. The turnkey style have the wizard names in all capital letters, for example ORACLE and GENERIC. The resource style have the wizard names with first letter capitalized, for example Oracle and Ippaddress.



For information on the availability of the RMS Application Wizards, contact your local customer support service or refer to the RMS Wizards documentation package (refer to Section “Further reading” for more details).

### 3.2.1.1 Turnkey wizards

The main function of turnkey wizards is that they deal with monitoring objects only, without regard to the internal RMS representation as a graph. This is their main advantage. The GENERIC turnkey wizard monitors almost every basic operating system object like file systems or IP address, and you do not need to be concerned with the proper linkage of sub-applications. This kind of wizard is written in capitals only (for example, DEMO, GENERIC).

### 3.2.1.2 Resource wizards

The subordinate wizards that the turnkey wizards invoke are mainly resource wizards; that is, resource-oriented wizards to configure certain kinds of resources like file systems or IP addresses.

Unlike the turnkey wizards, which have their names written with all capital letters (for example, DEMO, GENERIC), the resource wizards begin with an initial capital letter only.

The following are some of the more important resource wizards:

- **Cmdline**—Configures the starting, stopping, and checking of resources by means of corresponding command lines.
- **Fsystem**—Configures local or remote file systems, preferably in a clustered environment.
- **Ipaddress**—Configures the IP addresses that are needed for communication by means of a LAN interface.
- **Rcvm**—Configures disk groups in a clustered environment administrated by the PRIMECLUSTER Volume Manager (not available in all areas).
- **Vxvm**—Configures disk groups in a cluster environment administrated by the VERITAS volume manager (not available in all areas).
- **Controller**—Configures applications that are in control of other applications.
- **Gds**—Configures disk class of PRIMECLUSTER GDS using RMS environment.

## 3.2.2 Site preparation

You will have to perform some preliminary site preparation before using the RMS Wizards to set up a high-availability configuration for an application. Site preparation ensures that all the resources scheduled for use in the high-availability configuration are actually available. This means, for instance, that the system files into which resources are entered may have to be modified.

The resources that may have to get their entries modified include, above all, file systems and LAN interfaces. For an example of entries made in these files, refer to the RMS Wizards documentation package (refer to the Section “Further reading” for more details).

### 3.2.2.1 NFS Lock Failover

NFS Lock Failover feature applies to local file systems. If you enable NFS Lock Failover and the file system subsequently fails, the NFS locks associated with the file system also fail over along with the file system. To take advantage of this feature, the following site preparation steps need to be taken:

- You must have a shared disk accessible to all nodes in the cluster.
- Internal implementation of NFS Lock Failover needs a dedicated directory. You need to specify a directory that does not already exist. The directory will be used solely for NFS Lock Failover. Therefore, if you specify a directory that already exists, no other applications will be allowed to use it thereafter. Specify the name of the directory using menu item *NFSLockFailover* in *Global Settings* (see Figure 3). The directory entered in this screen will be created on all shared file systems selected for NFS Lock Failover. For example, if the directory `nfs_lock_dir` is entered in this screen and the file system `/usr/test1` in `userApplication APP1` is selected for NFS Lock Failover, then a directory `/usr/test1/nfs_lock_dir` will be created (if it does not already exist) and will be used for storing lock information.

**i** Only one file system per `userApplication` object can be selected for NFS Lock Failover. For a more detailed description, refer to the HTML documentation for the `Fsystem` wizard.

```
Shared Directory for NFS Lock Failover:
Currently set: none
1) HELP
2) FREECHOICE
3) RETURN
Global setting: Enable NFS Lock Failover:
```

Figure 3: NFS Lock Failover screen

- The directory entered in this screen must be accessible to all the nodes in the cluster. Otherwise, NFS failover will not work.
- This directory is reserved for NFS Lock Failover only.



This directory must not be used by any other applications.

- If the directory entered by the user begins with a slash (/) character, this character is dropped before creating the `/usr/test1/nfs_lock_dir` directory.
- Reserve one IP address for each `userApplication` object from which all the local file systems (set with NFS Lock Failover) must be shared.

### 3.2.2.2 Important files

The most important files that may need to be modified are as follows:

- `/etc/vfstab`—Contains entries for all of the local file systems that are to be used as resources in the high-availability configuration. In other words, this file contains the entries of the file systems that need to be mounted locally.

The following is an example of an entry into the `/etc/vfstab` file:

```
#RMS#/dev/dsk/c0t0d0s0 /dev/rdk/c0t0d0s0 /testfs1 ufs 1 yes -
```

- `/etc/dfs/dfstab`—Contains entries for all of the shared remote resources in the high-availability configuration. In other words, this file contains the entries for the file systems that must be mounted on a remote node.

To configure Solaris systems, it is important to have at least one non-RMS entry in this file. This has to be an entry that can be read at boot time; that is, it must not have entered as a comment. Otherwise, reading the file would not cause any NFS daemon to be started.

This is because the RMS entries are normally entered as comments, which means that they will be ignored by all commands excepting those used by RMS or the RMS Wizards.

The non-RMS entry might be a dummy entry configured for a local file system and shared only to the local node. This would mean that no real sharing to a remote node is done, but it would still cause the NFS daemons to be started.

The following is an example of an entry in the `/etc/dfs/dfstab` file (a non-RMS entry and an RMS entry):

```
share -F nfs -o ro=localhost /var/opt/example
```

```
#RMS# share -F nfs -o rw, root=
fuji2RMS:fuji2:045nfs045dia1:045msg:fuji2RMS: /sapmnt/045
```

- /opt/SMAW/SMAWRrms/etc/hvipalias—Contains entries for all of the LAN interfaces that are to be used as resources in the high-availability configuration.

The entries must provide information on the names and netmasks that are required for the LAN communication. Optionally, there may also be some routing information.

The following is an example of an entry into the /opt/SMAW/SMAWRrms/etc/hvipalias file:

#uname -n	IfName	Interface(s)	Netmask	Routes
fuji2	045dia1	hme1	0xffffffff00	

- /etc/hosts—Should contain the names of all the nodes that are part of the cluster for the high-availability configuration. If Cluster Admin was used to configure CIP for RMS, then /etc/hosts will already contain the RMS SysNode names.

This information should be available locally on each node to avoid problems in case of a failure to get access to the DNS.

- /var/adm/messages—By default, all RMS messages go to both the system log, /var/adm/messages, and the RMS switchlog file. If you do not want to send messages to the system log, then set HV\_SYS\_LOG\_USE = 0 in the hvenv.local file. The default is 1.

### 3.2.3 Creating and editing a configuration

After you finish with site preparation, you can begin the configuration procedure. This can be done by creating a configuration through the wizard menu.

You can bring up an existing wizard configuration that is running actively on the cluster nodes. In this case, you might call up the configuration because it is to be modified using the RMS Wizards. On the other hand, you might want to use the RMS Wizards to set up a new configuration. The commands for starting the RMS Wizards are as follows:



- `hvw`—Runs RMS Wizard Tools using the last activated configuration stored in the file `/opt/SMAW/SMAWRrms/etc/CONFIG.rms`. If this file does not exist or activation is being done for the first time, RMS uses the default configuration, `config`. Please refer to the Chapter “Manual pages” for additional information.
- `hvw -n config_file`—Edits an existing configuration, or it creates a new configuration using the name that you specify.

The sample configuration used for demonstration purposes in this chapter shows how to set up a new configuration called `mydemo` using the DEMO turnkey wizard. This example would be called up as follows:

```
# hvw -n mydemo
```

## 3.2.4 Using the wizard menus

Menus help you to set up a high-availability configuration. These menus are displayed after a configuration has been called up using the `hvw` command. The menus offer a number of items for performing the activities that are part of the configuration procedure.

The simplest way to make use of the menu items is to proceed as follows:

- Create an application, providing it with the features that make it suitable for running within a high-availability configuration.
- Activate the high-availability configuration.

Together with some modifications to illustrate a more complex configuration, the Section “Sample configuration” demonstrates this procedure. The following sections detail how to use the menu options.

### 3.2.4.1 Main configuration menu

On their highest level, the options for setting up a high-availability configuration are presented in a menu that is displayed immediately after a configuration has been called up. The top-level menu changes dynamically at run time depending on whether RMS is running in the cluster and whether the configuration being edited is the current configuration. This menu is called the *Main configuration menu*.

The *Main configuration menu* shows the state of the RMS cluster by indicating either one the following:

- RMS is inactive
- Where RMS is up and running

If RMS is running anywhere in the cluster, any actions that could modify a running configuration are disabled and are not available. Additionally, the menu items that are available are modified such that no changes can be made to the running configuration.

When RMS is running and the configuration being edited is not the same as is running, the main menu is not restricted except that the *Configuration-Activate* menu option is not available.

### 3.2.4.2 Main configuration menu when RMS is inactive

If RMS is not running anywhere, then the entire top level menu is presented without restrictions. Figure 4 shows the *Main configuration menu* window when RMS is inactive.

```
fuji2: Main configuration menu, current configuration: mydemo
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 4: Main configuration menu when RMS is inactive

## Menu items

The *Main configuration menu* can perform the following activities when RMS not running anywhere in the cluster:

- *Application-Create*—Specifies which application to configure for high availability. In addition, this option specifies all the relevant settings for the application so that it can run in a high-availability configuration monitored by RMS. Among the most important of these settings is the name of the application and the selection of nodes where the application runs; that is, the cluster nodes that the application will be running on.

The user application should be configured to run on multiple nodes for a high-availability configuration.

The wizard assists you by supplying menus with basic and non-basic attributes, assigns values to the attributes, and prompts you if an attribute is mandatory.

By choosing the appropriate turnkey wizard for an application, the wizard will then provide predefined elements, like scripts and detectors, for the application in question. These elements have been developed especially for the respective type of application.

The wizard will also carry out consistency checks at certain stages of the configuration procedure in order to prevent inconsistent applications from running in a high-availability configuration.

- *Application-Edit*—Modifies an existing application.

An existing application can be modified using this menu item. The following modes are available for editing an application:

- Turnkey mode (highly recommended)—Turnkey mode is the default mode. This mode is highly recommended because it simplifies complicated tasks like creating linkages between application and subapplications.
  - Non Turnkey mode (only for expert users)—Non Turnkey mode is meant for advanced/expert users only. If this mode is to be used, some rules must be followed. Otherwise, the resulting configuration may remain in an inconsistent state and RMS will not start. Usage of this mode is not within the scope of this guide.
- *Application-Remove*—Removes an existing application from the high-availability configuration.

- *Application-Clone*—Clones an application. This feature is provided for users who want to create a new application that differs only slightly from an existing one. To do this, clone an application and modify only the parts that are necessary to create a new one.
- *Configuration-Generate*—Generates the following:
  - Runs consistency checks on the configuration
  - Creates RMS graph on the configuration in the form of the `.us` file

While this *Generate* option is being carried out, the wizard indicates the number of applications and resources that have been generated so far by displaying them as dots on the system screen.

The wizard provides a high-availability configuration that is ready to be activated. However, it does not activate the configuration. This is done with the *Activate* option, which is described next.

The *Activate* option includes the *Generate* option. However, sometimes you may want to generate a configuration only (for example, in a testing environment).


- *Configuration-Activate*—Generates and activates a high-availability configuration.

Selecting this item performs both the generation and activation parts of the configuration process in one step. For the details on the generation portion of the process, refer to the previous item of this list.


With this option, the wizard distributes the configuration data to the other nodes that are part of a high-availability cluster. Activating the configuration means that all the information is provided for RMS to run with the configuration (for example, by installing the relevant files).

While activating a configuration, not all needed machines will always be up. If one or more machines are not up at the time of activation, these machines need to be updated using the menu item *Configuration-Push*. This menu item is for updating one cluster node at a time and appears only if RMS is running.

- *Configuration-Copy*—Produces a copy of an existing configuration.
- *Configuration-Remove*—Removes an existing high-availability configuration.
- *Configuration-Freeze*—Prevents or allows a configuration to be changed. With this option, the configuration can be viewed, but not modified.

 *Configuration-Freeze* is password protected, which means that you must make up a password to use the freeze option and then enter it again to thaw the configuration.

- *Configuration-Thaw*—Releases the configuration from the frozen state.

 *Configuration-Thaw* is password protected, which means that you must enter the password used to activate the freeze option to thaw the configuration.

- *Configuration-Edit-Global-Settings*—Offers a variety of additional features. These features of a global nature are used to change either the configuration or the operation mode of the `hvw` command.
- *Configuration-Consistency-Report*—Provides a consistency check that verifies whether an application is running within a high-availability configuration and has actually been created using the configuration data provided by the respective wizard.

The wizard compares the currently activated wizard checksum against the wizard database checksum. One checksum is called the *LifeInfo*, the other is called the *BuildInfo*. If both checksums match for an application, it is certified that its running version conforms to what was configured by the wizard.

- *Configuration-ScriptExecution*—Allows administrators to run any script independent of RMS.

By selecting the resources configured for the application, the user can execute the scripts that are to bring the resources `Online` or `Offline`. To see the online scripts being executed, you can go through the resource list, which is displayed for this purpose, in ascending order. The return code indicates the proper functioning of the respective script.

- *RMS-CreateMachine*—Provides the capability to set up the list of nodes in the cluster. This list is the set of nodes to be used when activating a configuration. If a node is in the list, but not used in any application, then only the `SysNode` objects are created for that node. With this feature, you can create a very basic, `SysNode`-only configuration by setting up the list of nodes without any applications.
- *RMS-RemoveMachine*—Provides the capability to remove machines from the list of cluster hosts.

### 3.2.4.3 Main configuration menu when RMS is running

When RMS is running on the local node, the primary difference in the *Main configuration menu* is that the *Configuration-Push* menu item is visible instead of the *Configuration-Generate* menu item.

*Configuration-Push* provides the capability to update (push) the running configuration to another node that needs updating. For example, if one cluster node were down for maintenance, and you updated the RMS cluster configuration in the meantime, you could use *Configuration-Push* to update the node that was down for maintenance.

### 3.2.4.4 Secondary menus

Each of the main menu items has a number of secondary menus. These secondary menus perform tasks related to the main menu item. The secondary menus themselves can have sub-menus.

The *Application type selection* menu is an example of a secondary menu. You see this menu after selecting *Application-Create* from the main menu.

Figure 5 shows the *Application type selection* menu.



The list of application types listed in the secondary menu depends upon the option selected in number 4 *OPTIONS*. The possible values are *ShowAllAvailableWizards* and *ShowTurnkeyWizardsOnly*. The selection *ShowTurnkeyWizardsOnly* is the default shown in the example. The number and kind of wizards depends on the installed packages (for example, Wizard Kit, SIS).

```
Creation: Application type selection menu:
```

- 1) HELP
- 2) QUIT
- 3) RETURN
- 4) OPTIONS
- 5) DEMO
- 6) GENERIC

```
Application Type: 5
```

Figure 5: Application type selection

This option allows you to select an application type to be assigned to the application in question. This is an important step in the configuration procedure since it invokes the specific application-type wizard to provide all the predefined elements (for example, scripts and detectors) that go with that application type.

The Section “Sample configuration” shows how to use some of the secondary menus. A more detailed description of these menus is given in the RMS Wizards documentation package.

### 3.2.5 Basic and non-basic settings

When a high-availability configuration is set up using the RMS Wizards, a distinction is made between basic and non-basic settings. Both kinds of settings can be configured by the user by means of the wizard menus. Basic and non-basic settings are only devices designed to guide you safely through the configuration process, ensuring that all mandatory settings are configured. There is no real difference between basic and non-basic settings.

Among the basic settings is, for instance, the name that is specified for an application, or the name of the node that an application is to run on. For example, the basic settings appear after selecting the *DEMO* application type from the secondary menu as shown in the previous section.

Figure 6 shows the menu leading to the menu to configure basic settings.

```
Consistency check ...
Yet to do: process the basic settings using Machines+Basics
Yet to do: choose a proper application name

Settings of turnkey wizard "DEMO"
1) HELP
2) NO-SAVE+EXIT
3) SAVE+EXIT
4) REMOVE+EXIT
5) ApplicationName=APP1
6) BeingControlled=no
7) Machines+Basics(-)
Choose the setting to process: 7
```

Figure 6: Menu leading to basic settings

Figure 7 shows the menu to configure basic settings.

```
Consistency check ...

Machines+Basics (appl:consistent)
 1) HELP                               13) (FaultScript=)
 2) -                                   14) (AutoStartUp=no)
 3) SAVE+EXIT                           15) (AutoSwitchOver=No)
 4) REMOVE+EXIT                          16) (PreserveState=no)
 5) AdditionalMachine                    17) (PersistentFault=0)
 6) AdditionalConsole                    18) (ShutdownPriority=)
 7) Machines[0]=fuji2RMS                 19) (OnlinePriority=)
 8) (PreCheckScript=)                   20) (StandbyTransitions=)
 9) (PreOnlineScript=)                  21) (LicenseToKill=no)
10) (PostOnlineScript=)                  22) (AutoBreak=yes)
11) (PreOfflineScript=)                  23) (HaltFlag=no)
12) (OfflineDoneScript=)

Choose the setting to process:
```

Figure 7: Menu to configure basic settings



Figure 8 shows the menu that includes the non-basic settings.

Non-basic settings include settings that are configured for resources like file systems, IP addresses, disks, and so forth. This menu appears after you complete the configuration of the basic settings.

Items enclosed in parenthesis are optional.



The menu to configure non-basic settings also includes the items provided for the basic settings.

```

Yet to do: process at least one of the non-basic settings

Settings of turnkey wizard "DEMO"
1) HELP                                9) DEMO(-)
2) -                                    10) LocalFileSystems(-)
3) SAVE+EXIT                            11) RemoteFileSystems(-)
4) -                                    12) IpAddresses(-)
5) ApplicationName=APP1                 13) RawDisks(-)
6) Machines+Basics(app1)               14) RC-VolumeManagement(-)
7) CommandLines(-)                    15) VERITAS-VolumeManagement(-)
8) Controllers(-)

Choose the setting to process:

```

Figure 8: Menu to configure non-basic settings

### 3.2.6 Usability

The wizard menus guide you in a way designed to be self-explanatory.

The following are some of the most frequently used menu items:

- *Selecting items*—This is normally done by typing the number of the item to be selected. Within the menu, a prompting line indicates the kind of input that is required.
- *Responding to messages*—Within the menus, several kinds of messages are displayed. One type of message might be to inform the user about the activities that the wizard has performed; for example, a consistency check that ended in a positive result. Other messages may prompt the user to continue the configuration procedure with a certain activity; for example, choosing an application name.

- *HELP*—This item provides user assistance and is available at the top of every wizard menu.
- *QUIT*—This is for leaving the wizard menu system.
- *RETURN*—This is for moving one level upwards in the menu system; that is, from a subordinate menu to the menu that it was called up from.
- *SAVE+EXIT/NOSAVE+EXIT*—This is for saving your input and for exiting. You can choose either to save the input or to discard it.

## 3.2.7 Configuration elements

This section deals with some basic elements that are part of a high-availability configuration. Most of them have been mentioned in previous sections; however, more details are provided here to assist you in understanding the RMS Wizards.



Users do not have to deal with any of the items listed in this section directly. RMS Wizards manage all the basic elements for a high availability configuration. This section is provided only to help users better understand the configuration elements.

### 3.2.7.1 Scripts

Scripts are used in a high-availability configuration to perform several kinds of actions. Among the most important types of actions are the following:

- Bringing a resource to an `Online` state
- Bringing a resource to an `Offline` state

As an example of a script bringing a resource `Offline`, you might think of a file system that has to be unmounted from a node where a fault occurs. An offline script will then use the `umount` command to unmount the file system. Another script might use the `mount` command to mount it on a new node.

Besides such online and offline scripts, there are also pre-online and pre-offline scripts for preparing transition into the respective states, as well as a number of other scripts.

The RMS Wizards, then, provide scripts that have been produced for applications matching a certain application type (for example, for R/3 applications). To configure high availability for an application, you assign the application to an application type, which means that all of the predefined scripts are then available for the configuration.



The `hvexec` command executes scripts for a high-availability configuration monitored by RMS. For more details on the command `hvexec` please refer to the *primer* document, which is part of the *RMS Wizards documentation package* (refer to the Section “Documentation”).

### 3.2.7.2 Detectors

Detectors are processes that have the task of monitoring resources. If there is a change in the state of a resource (for example, of a disk group) the detector in charge notifies the RMS base monitor. The base monitor may then decide to have a script executed as a reaction to this changed state.

The RMS Wizards provide detectors for use in a high-availability configuration. By assigning an application to an application type, the detectors become available for a configuration in the same way as was described for the scripts in the previous section.

### 3.2.7.3 RMS objects

A high-availability configuration can be seen as a set or group of objects with interdependencies. Any application or resource that is part of the configuration is then represented by one of the objects. The interdependences of objects can be displayed as a graph called the RMS graph.

A wizard is able to produce the RMS configuration based on the input you specified as well as on the predefined elements that are available to the wizard. The RMS configuration is created when you select the *Configuration—Generate* option, or the *Configuration—Activate* option, from the wizard main menu.

The difference in *Configuration—Activate* and *Configuration—Generate* is as follows:

- *Configuration—Generate* does not harm the running configuration. Generation is meant for checking changes to the configuration.
- *Configuration—Activate* overwrites the current RMS configuration data. The activation step must be carried out when RMS is not running, for example, this could be done at night if the application were to be shutdown for a short period.



This creates the configuration file stored in a file ending in `.us`, which is read by RMS at run-time.

There are several types of objects that are used in a configuration. In a high-availability configuration, objects of the same type can be monitored by one detector.

Among the more important object types are the following:

- `userApplication`—Represents an application to be configured for high-availability.
- `SysNode`—Represents a machine that is running as a node in a cluster.
- `gResource`—Represents a generic resource that is to be defined according to the needs of a the customer application.

## 3.3 Sample configuration

In going through this sample configuration, it is assumed that you are going to perform all of the configuration activities described in this section.

The first step is to create a configuration using the DEMO turnkey wizard.

In the menus in the sections that follow, you will be instructed on how to edit the configuration using the command `hvw -n config_file`.

`config_file` is the name of your RMS configuration file. The RMS configuration file name and path should not exceed 80 characters.

### 3.3.1 Creating an application

Enter the following command to generate the wizard menu for the sample configuration, `mydemo`:

```
# hvw -n mydemo
```

Before creating an application, set up the cluster hosts by selecting the nodes for the configuration.

The *RMS-CreateMachine* menu displays the current set of nodes and provides a list of machines that can be selected. If you select menu item 5) *ALL-CF-HOSTS*, the RMS Wizards selects all the hosts from `/etc/cip.cf` to be added to the host list of this configuration. Otherwise, hosts from the available list can be added one by one.

```
fuji2: Main configuration menu, current configuration: mydemo
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: 16
```

Figure 9: Main configuration menu

- ▶ Select the *RMS-CreateMachine* item by typing the number 16 (see Figure 9).

```
Creation: Add hosts to a cluster:
Current set: fuji2RMS
 1) HELP
 2) QUIT
 3) RETURN
 4) FREECHOICE
 5) ALL-CF-HOSTS
 6) fuji2RMS
 7) fuji3RMS
Choose the host to add: 7
```

Figure 10: RMS-CreateMachine menu

- ▶ Select *fuji2RMS* by typing the number 6 and select *fuji3RMS* by typing the number 7 (see Figure 10).
- ▶ Select *RETURN* by typing the number 3 to go back to the RMS Wizards *Main configuration menu*.

Machines can be removed from the list of cluster hosts by using menu item *RMS-RemoveMachine* from the *Main configuration menu* (see Figure 9). The menu contains all of the currently set cluster nodes (see Figure 11).

```
Removal: Remove hosts from a cluster:
Current set: fuji2RMS fuji3RMS
 1) HELP
 2) QUIT
 3) RETURN
 4) ALL
 5) fuji2RMS
 6) fuji3RMS
Choose the host to remove:
```

Figure 11: RMS-RemoveMachine menu

Machines can be removed by selecting them individually or by selecting 4) *All* from the menu. If you select 4) *ALL*, all the machines are removed; however, machines being used by one or more `userApplication` objects cannot be removed.

The next sample configuration procedure is to define an application that is to be highly available. To perform this first step, use the *Application-Create* option (see Figure 12).

```
fuji2: Main configuration menu, current configuration: mydemo
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: 3
```

Figure 12: Main configuration menu

- ▶ Select the *Application-Create* item by typing the number 3.

Next, the *Application type selection* menu allows you to assign an application type to the application that you want to configure (see Figure 13).



The list of application types listed in the secondary menu depends upon the option selected in number 4) *OPTIONS*. The possible values are *ShowAllAvailableWizard* and *ShowTurnkeyWizardsOnly*. The selection *ShowTurnkeyWizardsOnly* is the default shown in the example. The number and kind of wizards depends on the installed packages (Wizard Kit and additional market-specific packages).

```
Creation: Application type selection menu:  
1) HELP  
2) QUIT  
3) RETURN  
4) OPTIONS  
5) DEMO  
6) GENERIC  
Application Type: 5
```

Figure 13: Application type selection menu

This sample configuration uses the *DEMO* type of application to configure.

- ▶ Select the *DEMO* application type by typing the number 5.

You have now assigned the *DEMO* application type to your application. This means the *DEMO* turnkey wizard will provide the application with scripts and detectors that were developed for this application type.

There are, however, some more features that need to be specified in order to create a high-availability application. One of these features is the name of the node to which the failover will be performed in case there is a fault with the node on which the application is running.

Another feature might be the application name; that is, the name of the application that is being configured to become a high-availability application. There is no need, however, to specify an application name in this sample configuration since the *DEMO* wizard provides *APPI* as a default here.

The *DEMO* wizard (*APPI*) is a simple application that displays an animated graphical figure on your screen, which was developed for this sample configuration. This application was designed to show the starting, stopping, and switching of the corresponding application. Furthermore, the procedure of killing the processes clearly shows the auto-recover capabilities of RMS.

After performing a consistency check, the wizard informs you what to do next (see Figure 14).

```
Consistency check ...
Yet to do: process the basic settings using Machines+Basics
Yet to do: choose a proper application name

Settings of turnkey wizard "DEMO"
1) HELP
2) NO-SAVE+EXIT
3) SAVE+EXIT
4) REMOVE+EXIT
5) ApplicationName=APP1
6) BeingControlled=no
7) Machines+Basics(-)
Choose the setting to process: 7
```

Figure 14: Prompting for further actions

- ▶ Select the *Machines+Basics* item by typing the number 7.

If you want to specify a different application name, you could do it here by selecting item 5). However, because we are using the default of APP1, the *Yet to do* prompt will disappear after you select *Machine+Basics*.

The consistency of the application in question is checked anyway by the wizard to ensure that only consistent applications are allowed to be part of the high-availability configuration.



Selecting the *Machines+Basics* item takes you to a menu where, besides other features, you can specify an additional node. This is the node on which your application will run in case the current node or the application faults (see Figure 15).

```
Consistency check ...

Machines+Basics (appl:consistent)
1) HELP                               13) (FaultScript=)
2) -                                   14) (AutoStartUp=no)
3) SAVE+EXIT                           15) (AutoSwitchOver=No)
4) REMOVE+EXIT                         16) (PreserveState=no)
5) AdditionalMachine                   17) (PersistentFault=0)
6) AdditionalConsole                   18) (ShutdownPriority=)
7) Machines[0]=fuji2RMS                19) (OnlinePriority=)
8) (PreCheckScript=)                  20) (StandbyTransitions=)
9) (PreOnlineScript=)                 21) (LicenseToKill=no)
10) (PostOnlineScript=)                22) (AutoBreak=yes)
11) (PreOfflineScript=)                23) (HaltFlag=no)
12) (OfflineDoneScript=)

Choose the setting to process: 5
```

Figure 15: Consistency check and Machines+Basics menu

In addition to displaying the *Machines+Basics* menu, the wizard now also shows you the result of the latest consistency check. The application named *APPI*, which was indicated on the previous screen, has proven to be consistent.

Furthermore, under *Machines[0]* is indicated the node that your application is configured for at the moment. It is *fuji2RMS*.

You can now proceed to configure a failover node for your application as follows:

- ▶ Select the *AdditionalMachine* item by typing the number 5.

This takes you to a menu containing a list of nodes to choose from (see Figure 16).

```
1) HELP
2) RETURN
3) fuji2RMS
4) fuji3RMS
Choose a machine for this application: 4
```

Figure 16: List of nodes for failover procedure



The RMS Wizards retrieve the default list of nodes from the CF configuration files.

Your application is presently configured for *fuji2RMS*. *fuji3RMS* should then become the additional node:

- ▶ Select *fuji3RMS* by typing the number 4.

In the screen that follows you will see your selection confirmed. Under *Machines[1]* is now displayed *fuji3RMS* as the additional node.

This means that your application is configured to switch over to `fujj3RMS` if there is a failure on `fujj2RMS`. However, to actually switch your application over, manual action is required since `No` is specified as default for the `AutoSwitchOver` item in the `Machine+Basics` menu (see Figure 17).

```
Consistency check ...

Machines+Basics (appl:consistent)

1) HELP                               13) (OfflineDoneScript=)
2) -                                   14) (FaultScript=)
3) SAVE+EXIT                           15) (AutoStartUp=no)
4) REMOVE+EXIT                         16) (AutoSwitchOver=No)
5) AdditionalMachine                   17) (PreserveState=no)
6) AdditionalConsole                   18) (PersistentFault=0)
7) Machines[0]=fujj2RMS                19) (ShutdownPriority=)
8) Machines[1]=fujj3RMS                20) (OnlinePriority=)
9) (PreCheckScript=)                  21) (StandbyTransitions=)
10) (PreOnlineScript=)                 22) (LicenseToKill=no)
11) (PostOnlineScript=)                23) (AutoBreak=yes)
12) (PreOfflineScript=)                24) (HaltFlag=no)

Choose the setting to process: 16
```

Figure 17: Machines+Basics menu for additional nodes

To have the switchover procedure carried out automatically, you have to select the *AutoSwitchOver* item in this menu, and then specify the `AutoSwitchOver` mode from the menu that follows (see Figure 18).

```
Set flags for AutoSwitchOver: Currently set: NO (N)
1) HELP
2) -
3) SAVE+RETURN
4) DEFAULT
5) NO(N)
6) HOSTFAILURE(H)
7) RESOURCEFAILURE(R)
8) SHUTDOWN(S)
Choose one of the flags: 6
```

Figure 18: AutoSwitchOver mode

- ▶ Set a flag by entering the number 6 for `HOSTFAILURE`. This means that RMS switches an application to another node automatically in the case of a node failure.

```
Set flags for AutoSwitchOver: Currently set: HOSTFAILURE (H)
1) HELP
2) -
3) SAVE+RETURN
4) DEFAULT
5) NO(N)
6) NOT:HOSTFAILURE(H)
7) RESOURCEFAILURE(R)
8) SHUTDOWN(S)
Choose one of the flags: 7
```

Figure 19: Flags for AutoSwitchOver

- ▶ Enter the number 7 for `RESOURCEFAILURE` (see Figure 19). This means that RMS switches an application to another node automatically in the case of a resource failure.

This completes the *Application-Create* process. All the steps that are required to create a high-availability application have been taken.

Save your settings now (see Figure 20) and exit the first part of the configuration procedure as follows:

- ▶ Select *SAVE+EXIT* by typing the number 3.

```
Consistency check ...

Machines+Basics (appl:consistent)
 1) HELP
 2) -
 3) SAVE+EXIT
 4) REMOVE+EXIT
 5) AdditionalMachine
 6) AdditionalConsole
 7) Machines[0]=fujj2RMS
 8) Machines[1]=fujj3RMS
 9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
15) (AutoStartUp=no)
16) (AutoSwitchOver=HostFailure|ResourceFailure)
17) (PreserveState=no)
18) (PersistentFault=0)
19) (ShutdownPriority=)
20) (OnlinePriority=)
21) (StandbyTransitions=)
22) (LicenseToKill=no)
23) (AutoBreak=yes)
24) (HaltFlag=no)
Choose the setting to process: 3
```

Figure 20: Saving settings

You should, however, continue the configuration procedure by taking a look at the non-basic settings (see Figure 21).

```
Yet to do: process at least one of the non-basic settings

Settings of turnkey wizard "DEMO"
1) HELP                               9) DEMO(-)
2) -                                   10) LocalFileSystems(-)
3) SAVE+EXIT                           11) RemoteFileSystems(-)
4) -                                    12) IpAddresses(-)
5) ApplicationName=APP1                13) RawDisks(-)
6) Machines+Basics(app1)              14) RC-VolumeManagement(-)
7) CommandLines(-)                   15) VERITAS-VolumeManagement(-)
8) Controllers(-)

Choose the setting to process: 9
```

Figure 21: Non-basic settings

The screen following the *SAVE+EXIT* shows that another consistency check has been performed. This time the result is that for the present application there is at least one more setting that has still to be specified; however, it is not a basic option.

As was stated earlier, this application performs the task of displaying an animated graphical picture on an X-window display. Therefore, a display has to be specified for it in addition to the basic settings specified before.

In order to specify a display for the application that the *DEMO* wizard uses for demonstration, you have to select this wizard from the menu as follows:

- ▶ Select *DEMO* by typing the number 9.

```
Consistency check ...
Yet to do: set a display

CommandLines (Dem_APP1:not yet consistent)
1) HELP
2) -
3) SAVE+EXIT
4) REMOVE+EXIT
5) Display=
6) StartCommands[0]='hvexec~-F~demo~-c'
7) StopCommands[0]='hvexec~-F~demo~-u'
8) CheckCommands[0]=hvdet_demo
9) (Timeout=300)
10) (AutoRecover=no)
11) (MonitorOnly=no)
Choose the setting to process: 5
```

Figure 22: Prompting for display specification

On top of the screen, you see that a display still needs to be specified. Furthermore, you can see which scripts the wizard provides for starting, stopping, and checking. Their names are given in the lines beginning with *StartCommands[0]=*, *StopCommands[0]=*, and *CheckCommands[0]=*.



For technical reasons, tildes (~) are used for spaces within the wizard menu commands; however, the actual commands do not have tildes.

You are also told that *APP1* is not yet consistent; that is, *APP1* could not yet run within the *mydemo* configuration.

Specify the display within the *CommandLines* menu as follows:

- ▶ Select *Display* by typing the number 5.

You then get a list of display options (see Figure 23).

```
1) HELP
2) RETURN
3) FREECHOICE
4) fuji1ADM
5) fuji2ADM
6) fuji3ADM
7) fujiRCA
8) fujiSCON
9) fuji2
10) fuji3
11) fuji2RMS
12) fuji3RMS
Choose a display for this application: 3
      >> 172.25.220.27
```

Figure 23: List of display options

Selecting the *FREECHOICE* item enables you to make use of a display that is suitable for X-window type applications, which is what is required for *APPI*:

- ▶ Select *FREECHOICE* by typing the number 3.



On the screen is now the name of the node for this X-window type of display. Its name is the IP address: 172.25.220.27.

Selecting the *FREECHOICE* option causes another consistency check to be started (see Figure 24).

```
Consistency check ...

CommandLines (Dem_APP1:consistent)
1) HELP
2) -
3) SAVE+EXIT
4) REMOVE+EXIT
5) Display=172.25.220.27
6) StartCommands[0]='hvexec~-F~demo~-c~172.25.220.27'
7) StopCommands[0]='hvexec~-F~demo~-u~172.25.220.27'
8) CheckCommands[0]=hvdet_demo
9) (Timeout=300)
10) (AutoRecover=no)
11) (MonitorOnly=no)
Choose the setting to process: 3
```

Figure 24: Successful consistency check for APP1

The consistency check has successfully completed. This means that after the display is specified for *APP1*, it will run consistently within the *mydemo* configuration.



Specifying the display also led to a modification of the predefined scripts that the *DEMO* wizard provided.

This completes the specification of the non-basic settings, and at the same time, it completes the *Application-Create*. You can now save the non-basic settings, and exit this part of the configuration procedure (see Figure 24) as follows:

- ▶ Select *SAVE+EXIT* by typing the number 3.

This takes you back to the main menu for the settings of turnkey wizard *DEMO*, from which you can also exit (see Figure 25) as follows:

- ▶ Select *SAVE+EXIT* by typing the number 3.

```
Consistency check ...

Settings of turnkey wizard "DEMO"
1) HELP                      9) DEMO(Dem_APP1)
2) -                        10) LocalFileSystems(-)
3) SAVE+EXIT                 11) RemoteFileSystems(-)
4) -                        12) IpAddresses(-)
5) ApplicationName=APP1     13) RawDisks(-)
6) Machines+Basics(app1)   14) RC-VolumeManagement(-)
7) CommandLines(-)        15) VERITAS-VolumeManagement(-)
8) Controllers(-)

Choose the setting to process: 3
```

Figure 25: Turnkey wizard DEMO

By specifying the basic and non-basic settings for your application and by achieving a consistent result in the end, you have successfully finished the *Application-Create* part of the configuration procedure.

All that remains is to activate the *mydemo* configuration, so that the new application can run within it. The last exiting step will, therefore, take you back to the *Main configuration menu*. From here you can select *Configuration-Activate*.

### 3.3.2 Activating a configuration

Activating a configuration is the second of the two fundamental steps to set up a high-availability configuration. The activation step comprises a number of sub-steps, among which are generation and distribution of a configuration. The generation step can also be performed separately.

The starting point for the activating step is the *Main configuration menu* (see Figure 26).

```
fuji2: Main configuration menu, current configuration: mydemo
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: 8
```

Figure 26: Main configuration menu

- Select the *Configuration-Activate* item by typing the number 8.

The activation is performed by the wizard. No further input is required at this stage.

While the activation is performed, the wizard displays a number of sub-steps that belong to this part of the configuration procedure. The completion of a sub-step is indicated by displaying the word *done* or a similar expression (see Figure 27).

```
About to activate the configuration mydemo ...

Testing for RMS to be up somewhere in the cluster ... done

Arranging sub applications topologically ... done.

Check for all applications being consistent ... done.

Running overall consistency check ... done.

Generating pseudo code [one dot per (sub) application]: ... done

Generating RMS resources..... done

hvbuild using /usr/opt/reliant/build/wizard.d/mydemo/mydemo.us
About to distribute the new configuration data to hosts: fuji2RMS,fuji3RMS
The new configuration was distributed successfully.

About to put the new configuration in effect ... done.

The activation was finished successfully.
Hit CR to continue
```

Figure 27: Activating a configuration

Among the sub-steps carried out by the *Configuration-Activate* are generation and distribution of the configuration. The `hvbuild` command performs a consistency check of the RMS graph created in the generation of the configuration. The `hvdist` command distributes the configuration.

The test to see whether RMS is up on one of the nodes in the cluster is required since activation cannot be performed if RMS is running. In this case, RMS would need to be shut down first.

**i** The nodes that are currently not running RMS will have the persistent status information removed during the *Configuration—Activate* process.

After the configuration has been activated successfully, you can return to the *Main configuration menu*. From there, you can quit the configuration procedure.

**i** The Section “Creating a second application” continues this sample configuration after the first quit with an example on setting up a second application.

- ▶ Select **[Enter]** to return to the *Main configuration menu* (see Figure 28).

```
fuji2: Main configuration menu, current configuration: mydemo
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: 3
```

Figure 28: Quitting the Main configuration menu

- ▶ Select *QUIT* by typing the number 2.

This ends this portion of the configuration process. The following section contains an example on how to add a second application.

### 3.3.3 Creating a second application

In this section, the *mydemo* configuration is expanded by adding a second application. This example application differs from the first because duplicate configuration procedures are skipped to simplify the example. However, in other parts of the procedure, new features add to the complexity of the *mydemo* configuration.

The second application differs from the first as follows:

- The application uses a new application type, *GENERIC*, instead of *DEMO*. This means that the *GENERIC* turnkey wizard will be demonstrated in this example.
- The application has a default name *APP2* and is configured as being in control of the first application (*APP1*), requiring that the new application needs a controller.

After quitting the *Main configuration menu*, resume the configuration procedure as follows:

```
# hvw -n mydemo
```

The *Main configuration menu* opens (see Figure 29).

```
fuji2: Main configuration menu, current configuration: mydemo
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: 3
```

Figure 29: Starting again with the Main configuration menu

From the *Main configuration menu*, select *Application-Create* as follows:

- ▶ Select the *Application-Create* item by typing the number 3.

**i** The list of application types listed in the secondary menu depends upon the option selected in number 4) *OPTIONS*. The possible values are *ShowAllAvailableWizard* and *ShowTurnkeyWizardsOnly*. The selection *ShowTurnkeyWizardsOnly* is the default shown in the example. The number and kind of wizards depends on the installed packages (Wizard Kit and additional market-specific packages).

The *Application type selection* menu opens (see Figure 30).

```
Creation: Application type selection menu:
1) HELP
2) QUIT
3) RETURN
4) OPTIONS
5) DEMO
6) GENERIC
7) ORACLE
8) R3ANY
9) R3CI
10) RTP
11) WEBSERVER
Application Type: 6
```

Figure 30: Application type selection menu

This time, assign the *GENERIC* application type to the application, which means that the *GENERIC* turnkey wizard will be in charge of the configuration procedure.

- ▶ Select the *GENERIC* application type by typing the number 6.

After the consistency check, you are prompted to configure the basic settings. *APP2* is the default value for the application name.



If you want to change the name, select item 5) *ApplicationName* (see Figure 31).

```
Consistency check ...
Yet to do: process the basic settings using Machines+Basics
Yet to do: choose a proper application name

Settings of turnkey wizard "GENERIC"
1) HELP
2) NO-SAVE+EXIT
3) SAVE+EXIT
4) REMOVE+EXIT
5) ApplicationName=APP2
6) BeingControlled=no
7) Machines+Basics(-)
Choose the setting to process: 7
```

Figure 31: Prompting for further specification

- ▶ Select the *Machines+Basics* item by typing the number 7.



The consistency of APP2 is checked, and the result is positive. When the *Machines+Basics* menu appears, it shows that APP2 is also configured to run on fuji2RMS (see Figure 32).

```
Consistency check ...

Machines+Basics (app2:consistent)
1) HELP                               13) (FaultScript=)
2) -                                   14) (AutoStartUp=no)
3) SAVE+EXIT                          15) (AutoSwitchOver=No)
4) REMOVE+EXIT                        16) (PreserveState=no)
5) AdditionalMachine                  17) (PersistentFault=0)
6) AdditionalConsole                  18) (ShutdownPriority=)
7) Machines[0]=fuji2RMS              19) (OnlinePriority=)
8) (PreCheckScript=)                 20) (StandbyTransitions=)
9) (PreOnlineScript=)                21) (LicenseToKill=no)
10) (PostOnlineScript=)              22) (AutoBreak=yes)
11) (PreOfflineScript=)              23) (HaltFlag=no)
12) (OfflineDoneScript=)

Choose the setting to process: 5
```

Figure 32: Machines+Basics menu

- ▶ Select the *AdditionalMachine* item by typing the number 5.

The additional node to be specified for the failover procedure is fuji3RMS, as with the former application (see Figure 33).

```
1) HELP
2) RETURN
3) fuji2RMS
4) fuji3RMS
Choose a machine for this application: 4
```

Figure 33: List of nodes for failover procedure

- ▶ Select *fuji3RMS* by typing the number 4.

In the screen that follows you see your selection confirmed. Under *Machines[1]*, *fuji3RMS* is now displayed as the additional node. The new application is configured to be switched over to this node if there is a failure on the part of *fuji2RMS* (see Figure 34).

```
Consistency check ...

Machines+Basics (app2:consistent)
1) HELP                               13) (OfflineDoneScript=)
2) -                                   14) (FaultScript=)
3) SAVE+EXIT                           15) (AutoStartUp=no)
4) REMOVE+EXIT                         16) (AutoSwitchOver=No)
5) AdditionalMachine                   17) (PreserveState=no)
6) AdditionalConsole                   18) (PersistentFault=0)
7) Machines[0]=fuji2RMS                19) (ShutdownPriority=)
8) Machines[1]=fuji3RMS                20) (OnlinePriority=)
9) (PreCheckScript=)                   21) (StandbyTransitions=)
10) (PreOnlineScript=)                  22) (LicenseToKill=no)
11) (PostOnlineScript=)                 23) (AutoBreak=yes)
12) (PreOfflineScript=)                 24) (HaltFlag=no)

Choose the setting to process: 3
```

Figure 34: Machines+Basics menu

Save your settings and exit this part of the configuration procedure as follows:

- ▶ Select *SAVE+EXIT* by typing the number 3.

The basic settings have been specified. However, this application still needs to be set up to control another application in the non-basic settings.

For this reason, a controller has to be specified, which is an action within the non-basic settings. In addition, the application being controlled must be specified. This step follows the specification of the controller.

The *SAVE+EXIT* takes you to the menu for the non-basic settings (see Figure 35).

```
Consistency check ...
Yet to do: process at least one of the non-basic settings

Settings of turnkey wizard "GENERIC"
1) HELP
2) -
3) SAVE+EXIT
4) -
5) ApplicationName=APP2
6) Machines+Basics(app2)
7) CommandLines(-)
8) Controllers(-)
9) LocalFileSystems(-)
10) RemoteFileSystems(-)
11) IpAddresses(-)
12) RawDisks(-)
13) RC-VolumeManagement(-)
14) VERITAS-VolumeManagement(-)
Choose the setting to process: 8
```

Figure 35: Non-basic settings

- ▶ Select *Controllers* by typing the number 8.

This takes you to the menu for assigning a controller to the application, which is yet to be specified (see Figure 36).

```
Consistency check ...
Yet to do: assign at least one application to control
Yet to do: configure at least one controlled application without the M flag

Settings of application type "Controller" (not yet consistent)
1) HELP
2) -
3) SAVE+EXIT
4) REMOVE+EXIT
5) ControlPolicy=FOLLOW
6) AdditionalAppToControl
7) (InParallel=)
8) (FaultScript=)
Choose the setting to process: 6
```

Figure 36: Assigning a controller

- ▶ Select *AdditionalAppToControl* by typing the number 6.

Once you specify a controller, the wizard needs to know which application to control. Therefore, it offers you a list from which to choose an application (see Figure 37).

```
1) HELP
2) RETURN
3) FREECHOICE
4) app1
Choose an application to control: 4
```

Figure 37: List of applications to be chosen as controlled applications

The controlled application is APP1, while APP2 is the controlling application. Choose the application to be controlled as follows:

- ▶ Select *APP1* by typing the number 4.

There are a number of flags that can be set for a controlled application; that is, an application that is monitored by another application by means of a controller.

The following screen shows how to set one or more of these flags (see Figure 38).

```
Set flags for (sub) application: appl
Currently set: AUTORECOVER,TIMEOUT (AT180)
1) HELP
2) -
3) SAVE+RETURN
4) DEFAULT
5) MONITORONLY(M)
6) NOT:AUTORECOVER(A)
7) TIMEOUT(T)
Choose one of the flags:
```

Figure 38: Menu for setting controller flags

In this sample configuration, the *A* (AUTORECOVER) flag has been set. The *A* flag means that if the controlled application becomes *Offline*, the controlling application tries to restart it. The AUTORECOVER menu item is now in the opposite state; that is, ready to be toggled to *NOT*.

The *T* flag limits the amount of time tolerated while bringing the controlled application *Online* (see Figure 39).

```
Set flags for (sub) application: appl
Currently set: AUTORECOVER,TIMEOUT (AT180)
1) HELP
2) -
3) SAVE+RETURN
4) DEFAULT
5) MONITORONLY(M)
6) NOT:AUTORECOVER(A)
7) TIMEOUT(T)
Choose one of the flags: 7
```

Figure 39: Controller timeout period

- Change the timeout period by entering 7.

```
1) HELP
2) RETURN
3) FREECHOICE
4) 180
Set an appropriate timeout: 3
  >> 150
```

Figure 40: Changing controller timeout period

- ▶ Select *FREECHOICE* by typing the number 3 and type 150 for the timeout period (see Figure 40).
- ▶ Select **Enter** to return to the menu for controller flags (see Figure 41).

```
Set flags for (sub) application: appl
Currently set: AUTORECOVER,TIMEOUT (AT150)
1) HELP
2) -
3) SAVE+RETURN
4) DEFAULT
5) MONITORONLY(M)
6) NOT:AUTORECOVER(A)
7) TIMEOUT(T)
Choose one of the flags: 3
```

Figure 41: Saving flags for controller

After completing the settings, save them and return to the *Controllers* menu as follows:

- ▶ Select *SAVE+RETURN* by typing the number 3.

In this menu, you see your settings confirmed under *Controllers*. The flags have been set for APP1 (see Figure 42).

```
Consistency check ...

Settings of application type "Controller" (consistent)
1) HELP
2) -
3) SAVE+EXIT
4) REMOVE+EXIT
5) ControlPolicy=FOLLOW
6) AdditionalAppToControl
7) Controllers[0]=AT150:app1
8) (InParallel=)
9) (FaultScript=)
Choose the setting to process: 3
```

Figure 42: Indication of flags set for controller

Another consistency check proves the settings to be consistent.

- ▶ Select *SAVE+EXIT* by typing the number 3.

This takes you back to the *GENERIC* menu (see Figure 43).

```
Consistency check ...

Settings of turnkey wizard "GENERIC"
1) HELP
2) -
3) SAVE+EXIT
4) -
5) ApplicationName=APP2
6) Machines+Basics(app2)
7) CommandLines(-)
8) Controllers(Ctl_APP2)
9) LocalFileSystems(-)
10) RemoteFileSystems(-)
11) IpAddresses(-)
12) RawDisks(-)
13) RC-VolumeManagement(-)
14) VERITAS-VolumeManagement(-)
Choose the setting to process: 3
```

Figure 43: Menu with settings for *GENERIC* turnkey wizard

In the *GENERIC* menu it is now displayed that a controller has been assigned for *APP2*.

- ▶ Select *SAVE+EXIT* by typing the number 3.



This takes you back to the *Main configuration menu* (see Figure 44).

```
fuji2: Main configuration menu, current configuration: mydemo
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 44: Main configuration menu

This completes the creation of the second application.

After returning to the *Main configuration menu*, activate the *mydemo* configuration for the second time. This has to be done because you have modified the configuration by adding another application.

### 3.3.4 Activating the configuration a second time

To activate the configuration, begin by selecting *Configuration-Activate* from the *Main configuration menu* (see Figure 45).

```
fuji2: Main configuration menu, current configuration: mydemo
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: 8
```

Figure 45: Main configuration menu

► Select *Configuration-Activate* by typing the number 8.

The activation is performed by the wizard. No further input is required at this stage.

While the activation is carried out, the wizard displays a number of the sub-steps that belong to this part of the configuration procedure. The completion of a sub-step is indicated by displaying the word *done* or similar expression (see Figure 46).

```
About to activate the configuration mydemo ...

Testing for RMS to be up somewhere in the cluster ... done

Arranging sub applications topologically ... done.

Check for all applications being consistent ... done.

Running overall consistency check ... done.

Generating pseudo code [one dot per (sub) application]: ..... done

Generating RMS resources..... do
ne

hvbuild using /usr/opt/reliant/build/wizard.d/mydemo/mydemo.us
About to distribute the new configuration data to hosts: fuji2RMS,fuji3RMS
The new configuration was distributed successfully.

About to put the new configuration in effect ... done.

The activation was finished successfully.
Hit CR to continue
```

Figure 46: Activating the configuration for the second time

Refer to the Section “Activating a configuration” for the meaning of the sub-steps displayed above.

After the configuration has been activated successfully, return to the *Main configuration menu* and quit the configuration procedure.

- ▶ Select **[Enter]** to return to the *Main configuration menu* (see Figure 47).

```
fuji2: Main configuration menu, current configuration: mydemo
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: 2
```

Figure 47: Return to Main configuration menu

- ▶ Select *QUIT* by typing the number 2.

This ends the sample configuration.

## 3.4 Further reading

The preceding sections were intended to make the reader familiar with some basic concepts and methods of the RMS Wizards. More information may be obtained from a number of documents that provide further reading on these tools and the way they are used.

### RMS Wizards documentation package

The RMS Wizards documentation package is available on the PRIME-CLUSTER CD-ROM.

It deals with topics like the configuration of file systems and IP addresses, or the different kinds of wizards, in more detail.

HTML files on these topics can be found in the following sections:

- *Primer*

This section provides an introduction to the RMS Wizards. It covers many of their features in more detail than has been possible in this chapter.

The Section “Site preparation” contains many examples of the system files that are to be modified when performing the task of site preparation.

- *Wizards*

This section provides information on individual wizards of all three kinds described in this chapter; that is, turnkey wizards, resource wizards, and other wizards, including the *generic* wizard.

- *Scripts and tools*

This section provides information on some scripts and tools that may be useful in setting up a high-availability configuration by means of the RMS Wizards, including the *gresources* sub-section which contains descriptions of a number of detectors. *Gresources* are defined as physical system resources.

- *Manual*

This section provides what information is available on the manual pages for some commands that are frequently used to configure high availability for an application using the RMS Wizards.

The `hvw` and the `hvexec` commands, which were also described in this chapter, are explained here in more detail.

**Manual pages**

Information on the commands that are used for configuration with the RMS Wizards may also be obtained by calling up the manual pages.

The manual pages are available, for instance, for the `hvw` and the `hvexec` commands, which were also described in this chapter.

## 4 Administration

This chapter describes PRIMECLUSTER administration using the Cluster Admin graphical user interface (GUI). In addition, some command-line interface (CLI) commands are discussed.

This chapter discusses the following:

- The Section “Overview” introduces PRIMECLUSTER administration by means of the Cluster Admin and the CLI.
- The Section “Using Cluster Admin” discusses how to use the RMS portion of the GUI.
- The Section “RMS procedures” describes how to Administer RMS using the GUI. It also contains CLI commands as a convenience for advanced users.

### 4.1 Overview

RMS administration can be done by means of the Cluster Admin GUI or by the CLI; however, it is recommended that you use the Cluster Admin GUI. The CLI should only be used by expert system administrators or in cases where a browser is not available. The following sections primarily describe the Cluster Admin GUI options. The CLI equivalents are provided in the RMS procedures section.

### 4.2 Using Cluster Admin

The following sections discuss how to use the RMS portion of the GUI.

#### 4.2.1 Starting Cluster Admin

Open the Java-enabled browser (use Internet Explorer 5.x or Netscape Navigator 4.x or higher versions) and enter the following URL in the *Address* location:

```
http://nodename:8081/Plugin.cgi
```

The *nodename* should be the name or IP address of the primary or secondary management server. For example, if a CF cluster called FUJII has *fujii2* and *fujii3* as its primary and secondary management servers, the URL would be either one of the following:

- <http://fujii2:8081/Plugin.cgi>
- <http://fujii3:8081/Plugin.cgi>

Figure 48 shows an example of a browser with the URL entered in the *Address* location. Windows desktop systems require the Java plug-in as specified in the *Installation Guide*. For details on the primary and secondary management servers, please refer to the *Installation Guide*.

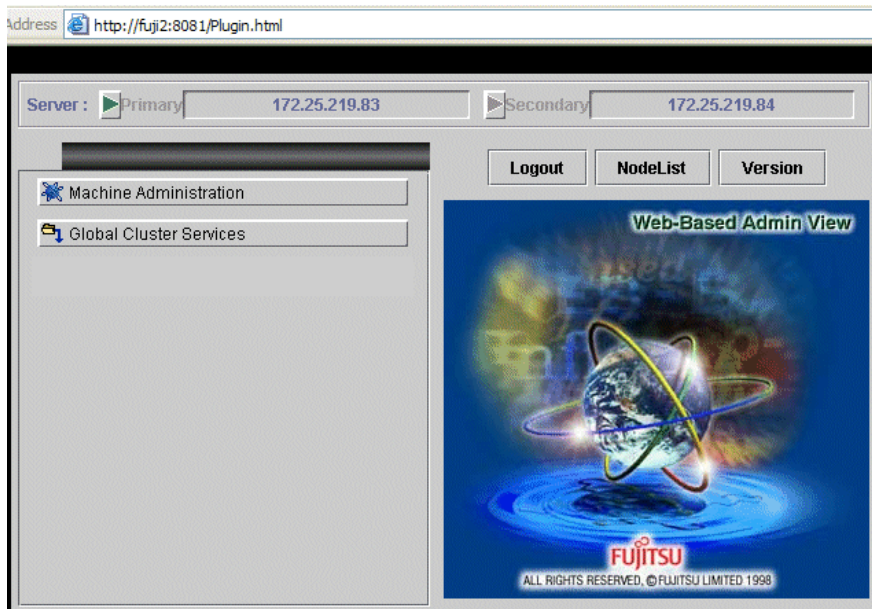


Figure 48: Invoking the Cluster Admin GUI



## 4.2.2 Logging in

After the Web-Based Admin View login screen appears (see Figure 49), log in as follows:

- ▶ Enter the user name and password for a user with the appropriate privilege level.
- ▶ Click on the *OK* button.

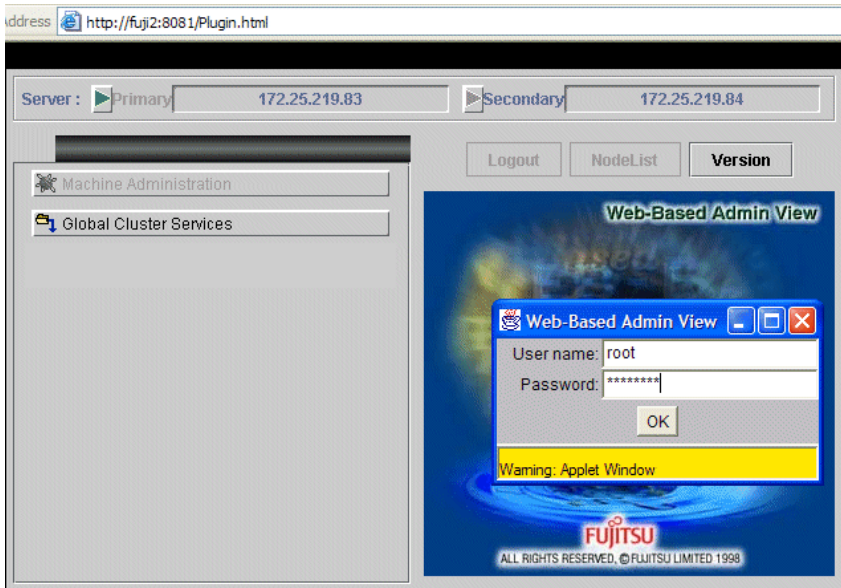


Figure 49: Web-Based Admin View login screen

Use the appropriate privilege level while logging in. Cluster Admin has the following privilege levels:

- Root privileges—Can perform all actions including configuration, administration, and viewing tasks.
- Administrative privileges—Can view and execute commands, but cannot make configuration changes.
- Operator privileges—Can only perform viewing tasks.

For more details on the privilege levels, refer to the *PRIMECLUSTER Installation Guide (Solaris)*.

After clicking on the *OK* button, the top menu appears (see Figure 50).



Figure 50: Top menu

Open Cluster Admin as follows:

1. Click on *Global Cluster Services*.
2. Click on the *Cluster Admin* button to start Cluster Admin
3. The *Choose a node for initial connection* screen appears (see Figure 51). Select a node, and click on *OK*. The main Cluster Admin screen appears.

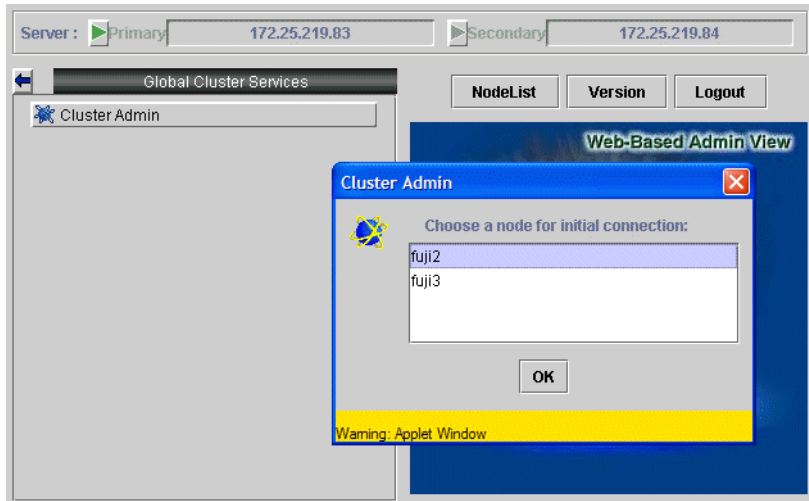


Figure 51: Cluster menu

### 4.2.3 Main screen

The main screen (see Figure 52) contains the following tabs on the left-hand side panel:

- *cf*
- *rms*
- *sis*
- *msg* (message window)

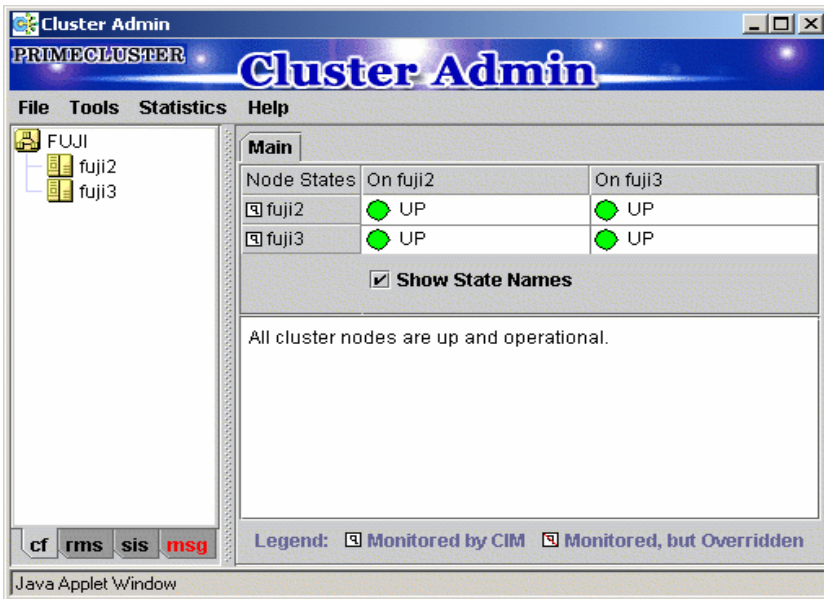


Figure 52: Main screen

Select the appropriate tab to switch to a component. By default, the *cf* tab is selected.

The Cluster Admin GUI has some standard components that are common across RMS, SIS, and CF. They are as follows:

- Pull-down menus—Pull-down menus that contain both functions generic to the Admin GUI and specific to the PRIMECLUSTER products.
- Tree panel—Panel on the left is normally the tree panel. This panel displays product-specific configuration information. Click on a tree component to view further information in the main panel.
- Main panel—Large panel on the right is the main work and information area. The content varies according to the product being administered and the functions selected from the menus or tree.

## 4.2.4 RMS main window

To start the RMS portion of the GUI, click on the *rms* tab. An example of the RMS main window is shown in Figure 53. The main window area is split into two sub-areas. The RMS tree is displayed on the left-hand side panel. The right-hand side panel is used to display configuration information or properties of nodes, logs, or both, depending on the selections in the RMS tree.

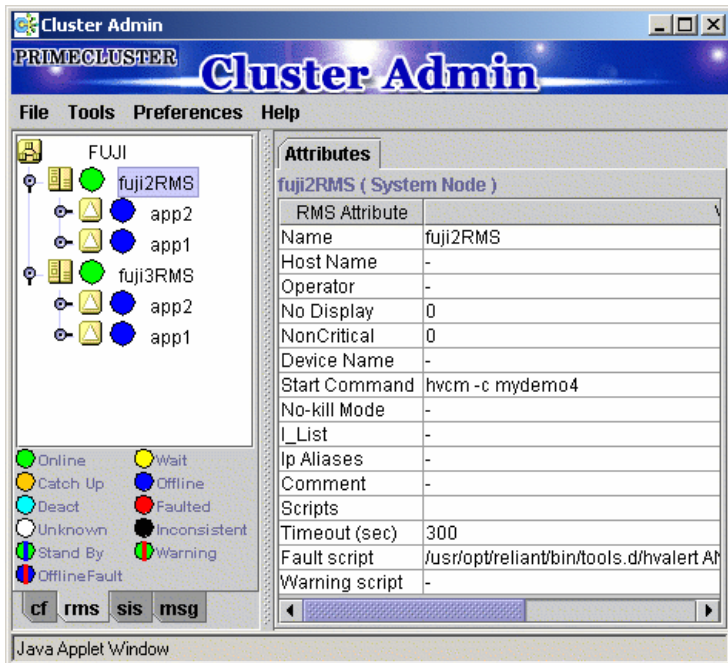


Figure 53: RMS main window

### 4.2.4.1 RMS tree

The RMS tree displays the configuration information of the cluster in a hierarchical format. The tree has the following levels:

- Root of the tree—Represents the cluster.
- First level—Represent the system nodes forming the cluster.

- Second level—Represent the `userApplication` objects running on each of the system nodes.
- Third level—Represent sub-applications, if any.
- Fourth level—Represents the resources necessary for each of the sub-applications.

If an application has sub-applications, the fourth level represents resources used by that sub-application. If an application does not have sub-applications, then the third level represents all the resources used by the `userApplication`.

Dependencies between the applications are depicted in the RMS tree by means of the controller object. An example of the RMS tree with a controller object is shown in Figure 54.

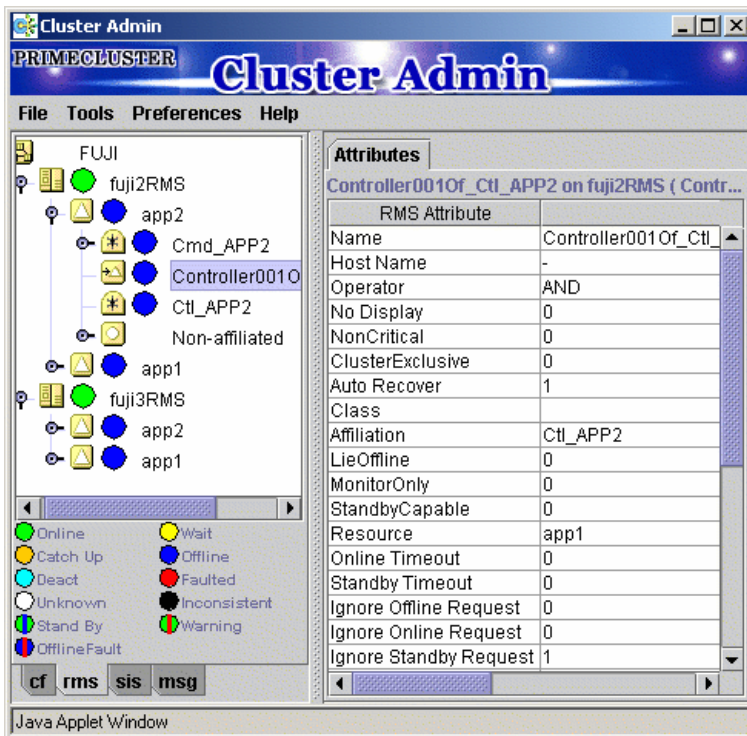


Figure 54: RMS tree with a controller object

#### 4.2.4.2 Configuration information or object attributes

View the configuration information for the individual objects by left-clicking with the mouse on the object in the tree. The properties are displayed in a tabular format on the right-hand side panel of the RMS main window (see Figure 55).

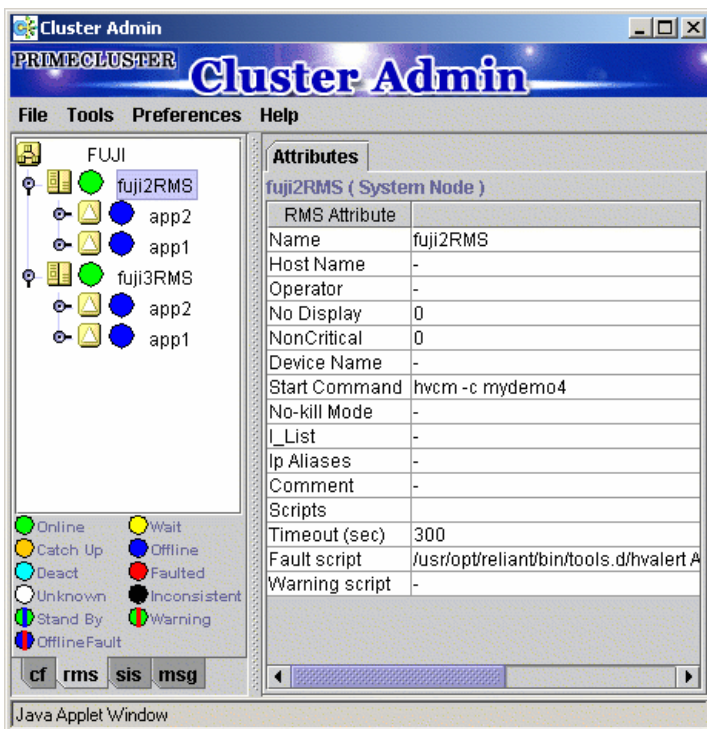


Figure 55: Configuration information or object attributes

#### 4.2.4.3 Command pop-ups

You can perform many operations on the RMS tree objects by using the context-sensitive command pop-up menus. Invoke the pop-up menu by right-clicking with the mouse on the object. The menu options are based on the type and the current state of the selected object (see Figure 56).



Figure 56: Command pop-up

For example, the menu offers different options for a `SysNode` object selection and `userApplication` object selection. It also offers different options for an `Online` `userApplication` object in the `Online` state and an `Offline` `userApplication` object in the `Offline` state (see Figure 57).





Figure 57: Command pop-up for an Offline application

#### 4.2.4.4 Switchlogs and application logs

The switchlog on individual system nodes can be viewed by using the *View Switchlog* option from the system node command pop-up window. The switchlog is displayed in a tab on the right-side panel (see Figure 58).

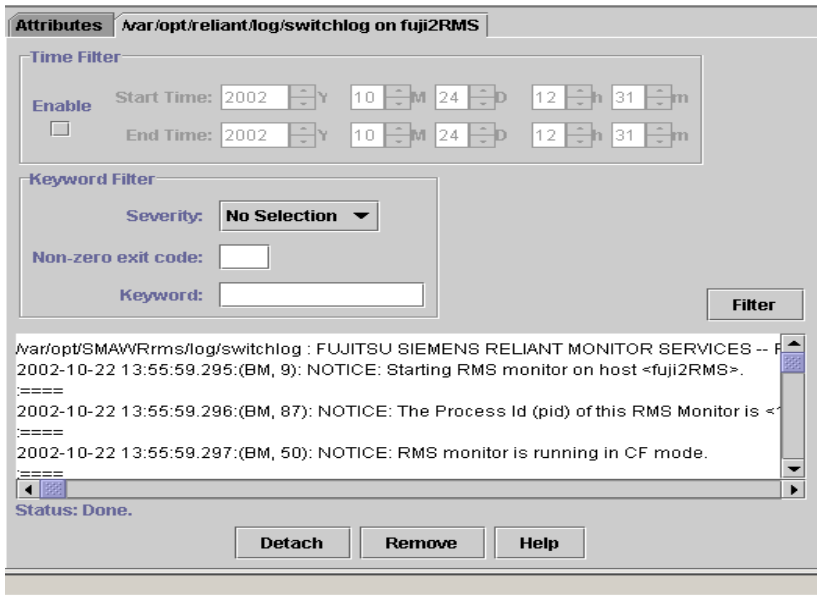


Figure 58: Viewing the RMS switchlog file

You can detach the switchlog panel from the main window and view it in a separate frame (see Figure 59). Additionally, the detached window can be attached back to the main window.

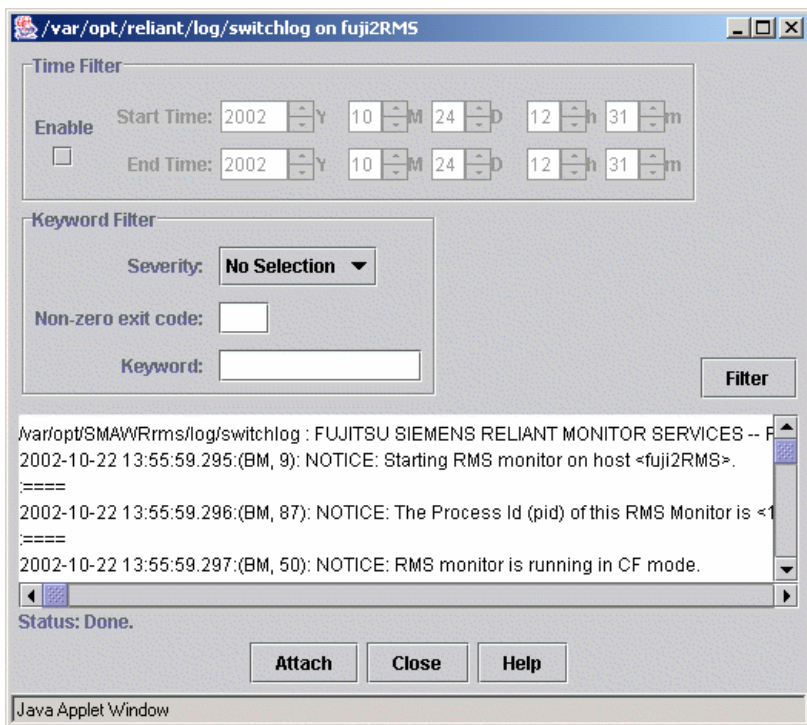


Figure 59: Viewing the RMS switchlog file in a detached window

Display the application log by right-clicking on an *Online* application on the RMS tree and choosing *View Logfile* (see Figure 60).

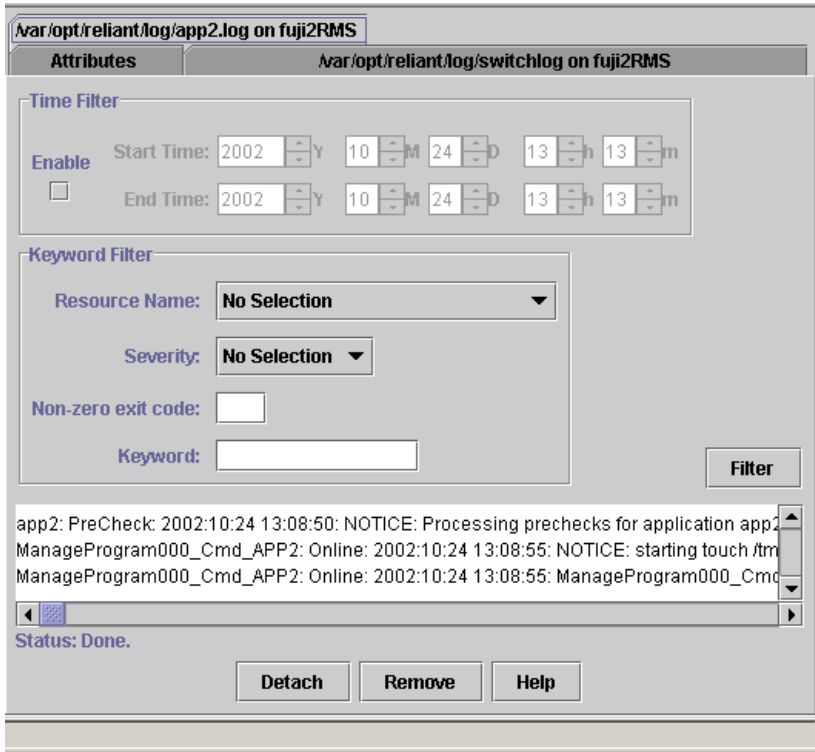


Figure 60: Viewing the application log

You can view the log files as follows:

- In their entirety
- Filter them by date or keywords
- Scroll through or jump to any entry
- Filter out irrelevant entries using search criteria based on keyword, date ranges, severity of error messages, and non-zero exit codes



Refer to the Chapter “Troubleshooting” for the description of severity levels and exit codes.

Figure 61 shows the screen for a search based on the date and time.

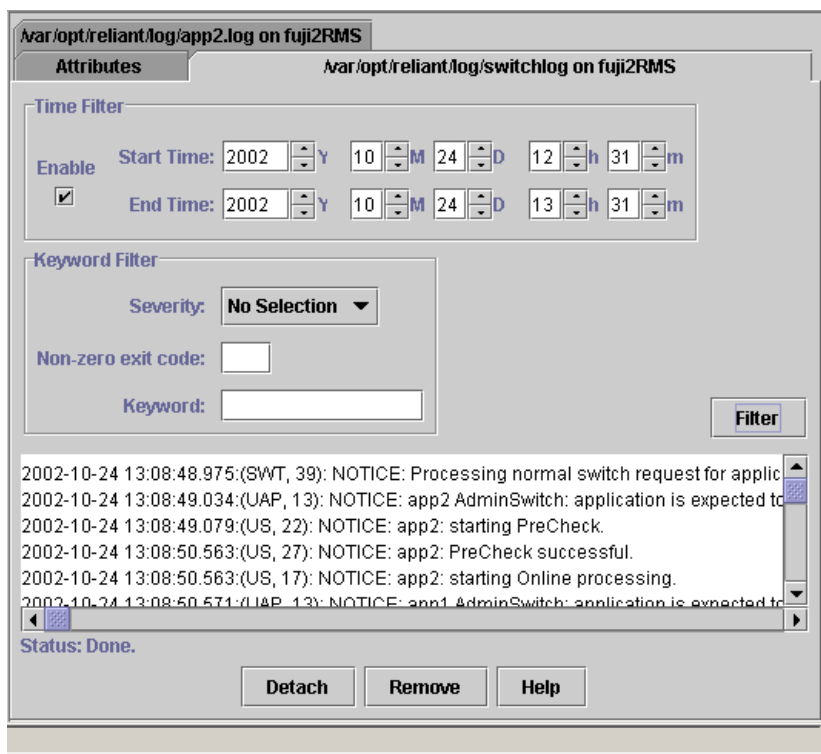


Figure 61: Search based on date and time filter

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* option (see Figure 62).

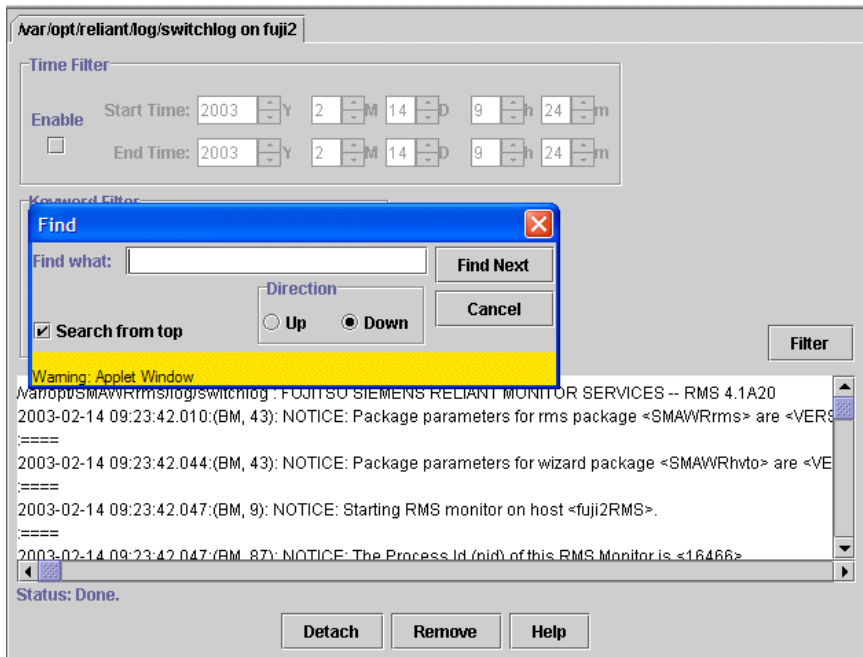


Figure 62: Using the Find pop-up in log viewer

## 4.2.5 RMS graphs

Cluster Admin contains the following RMS graphs, which are useful for graphically viewing the details of the RMS configuration file:

- Full graph—Displays the complete cluster configuration.
- Application graph—Shows all of the resources used by an application and can be used to look at specific resource properties.
- Sub-application graph—Lists all of the sub-applications used by a given application, and it shows the connections between the sub-applications.
- Composite sub-applications graph—Shows all the sub-applications that the application depends on directly or indirectly.

These graphs are explained in more detail in the sections that follow.

### 4.2.5.1 RMS full graph

The RMS full graph displays the complete configuration of the cluster (see Figure 63). The graph represents the following items in the RMS configuration:

- Relationships between objects
- Dependencies of objects
- Object types
- Current node state

You can see the RMS full graph by right-clicking on a system node. The RMS graph is drawn from the perspective of a particular system node; that is, the state information of all the nodes is displayed as viewed from a particular system node. You can view an RMS graph from the perspective of any of the system nodes. The node name in the title bar of the graph identifies the node that is supplying the state information.

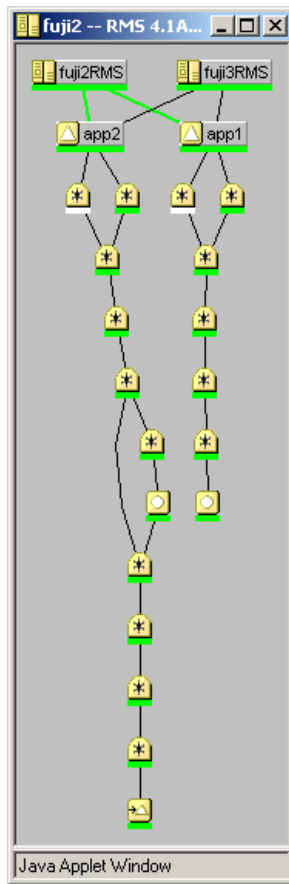


Figure 63: RMS full graph



### 4.2.5.2 Application graph

You can see a graph for a single application by right-clicking on an application. The application graph shows all the resources used by that specific application. You can also look at specific resource properties. The application graph is similar to the full graph, except that it shows just a single application and its resources. The graph is shown from the perspective of the selected node (see Figure 64).

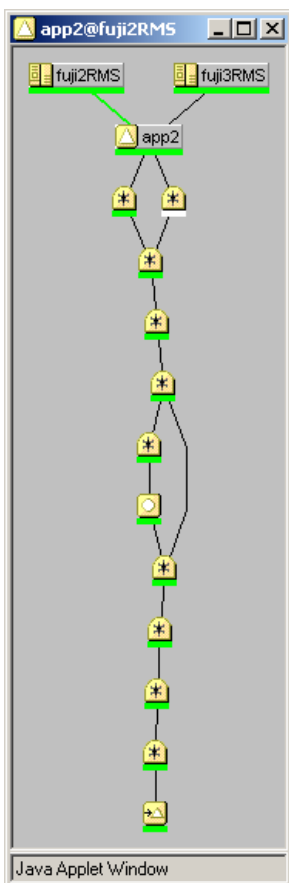


Figure 64: RMS application graph

### 4.2.5.3 Sub-application graph

You can see a graph for a sub-application by right-clicking on a sub-application. The sub-application graph lists all the sub-applications used by a given application, and it shows the connections between the sub-applications (see Figure 65).

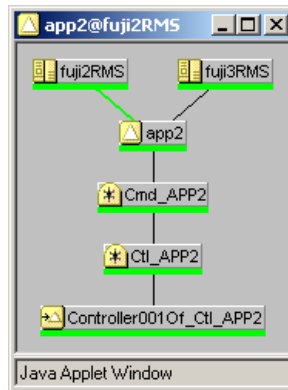


Figure 65: RMS sub-application graph

#### 4.2.5.4 Composite sub-application graph

The composite sub-application graph is a variation of the sub-application graph. If an application has a dependency on another application by means of a controller object, then the composite sub-application graph can be used to show all the sub-applications that the application depends on directly or indirectly. For example, a composite graph may depict a Web Server application that depends on a Oracle Database Server application.

The composite sub-application graph takes the controller object in a sub-application graph and appends the sub-application graph of the controlled application below it. This gives a composite view of all the sub-applications that the first application depended on directly or indirectly (see Figure 66). If the controlled application has further controller objects, then the process is recursively repeated.

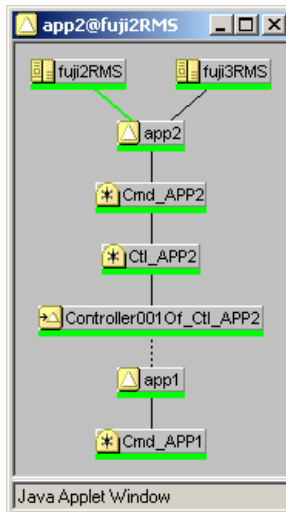


Figure 66: Composite sub-application graph

#### 4.2.5.5 Configuration information and the graphs

Click the left mouse button on the object of interest to see the configuration information of the object in a graph form. A pop-up screen displays the attributes (see Figure 67).

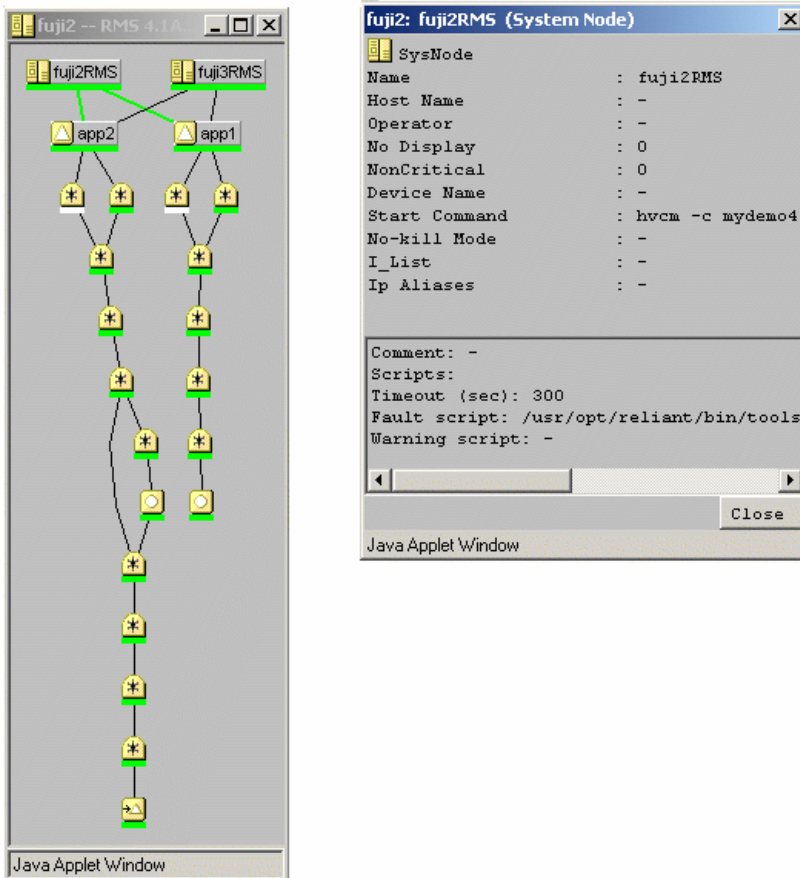


Figure 67: Configuration information pop-up

#### 4.2.5.6 Command pop-ups

You can use the context sensitive command pop-up menus on the RMS graph nodes to perform many operations. Invoke the pop-up menu by right-clicking on an object. The menu options are based on the type and the current state of the selected object (see Figure 68).

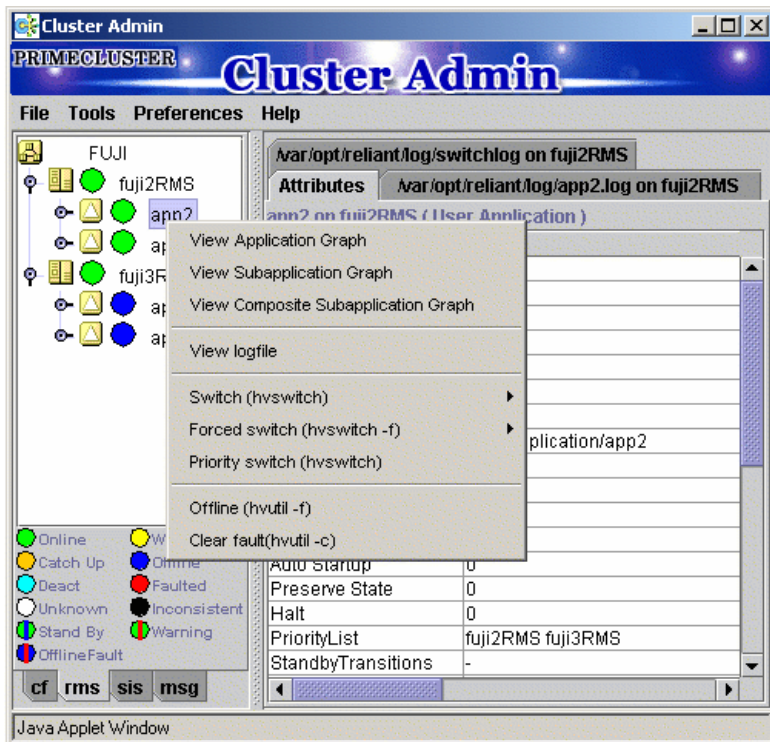


Figure 68: Command pop-up

#### 4.2.5.7 RMS graph customization

By default, the RMS graph does not display the resource (object) names for the full and application graphs. These are available as tool tips and can be seen by placing the mouse on a particular object. A menu option is provided from the *Preferences* menu to display the resource names on these graphs. Another menu option is provided to display the affiliation names (see Figure 69).

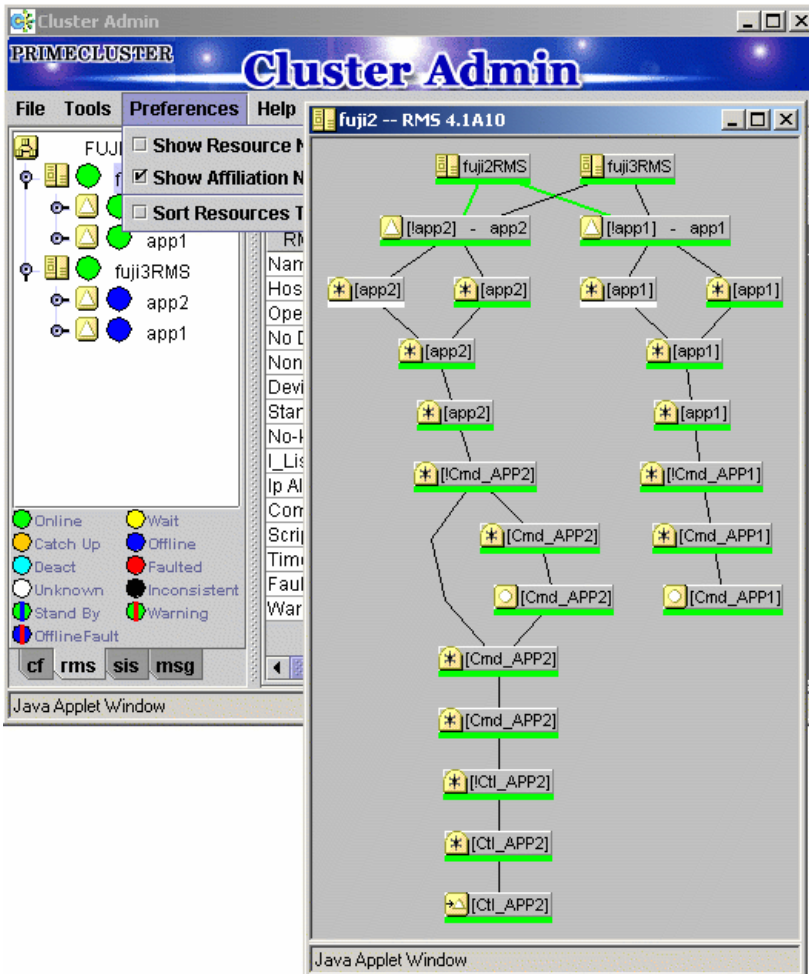


Figure 69: Graph with affiliations

If both the options are selected, graphs will be shown with both the affiliations and names for the resources (see Figure 70).

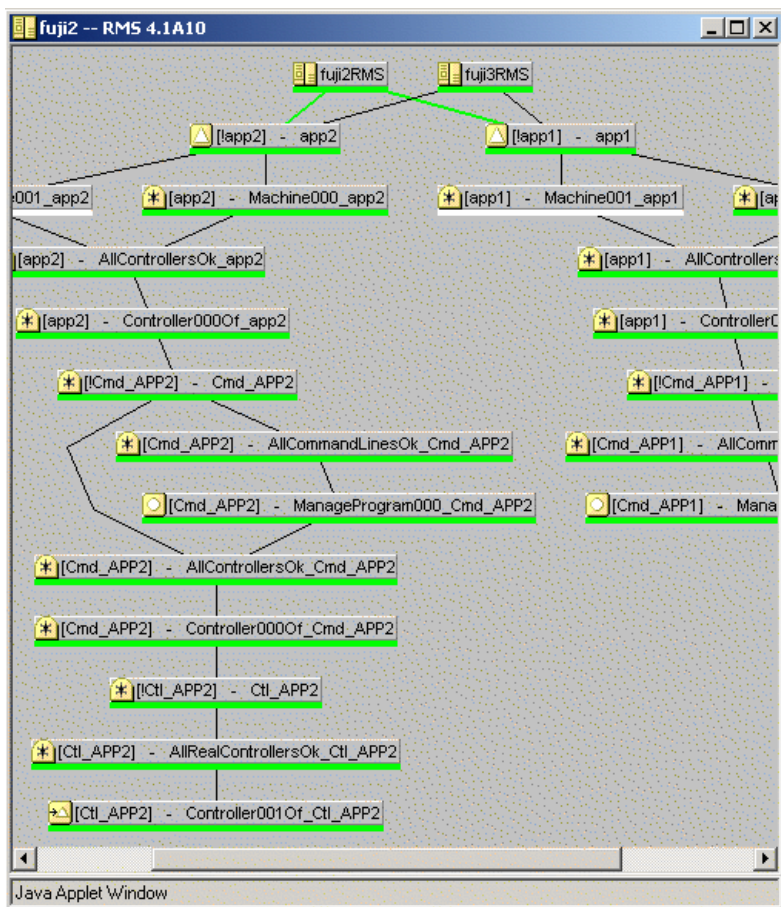


Figure 70: Graphs with affiliations and names

This option stretches the graph horizontally and can make it difficult to read (see Figure 71).

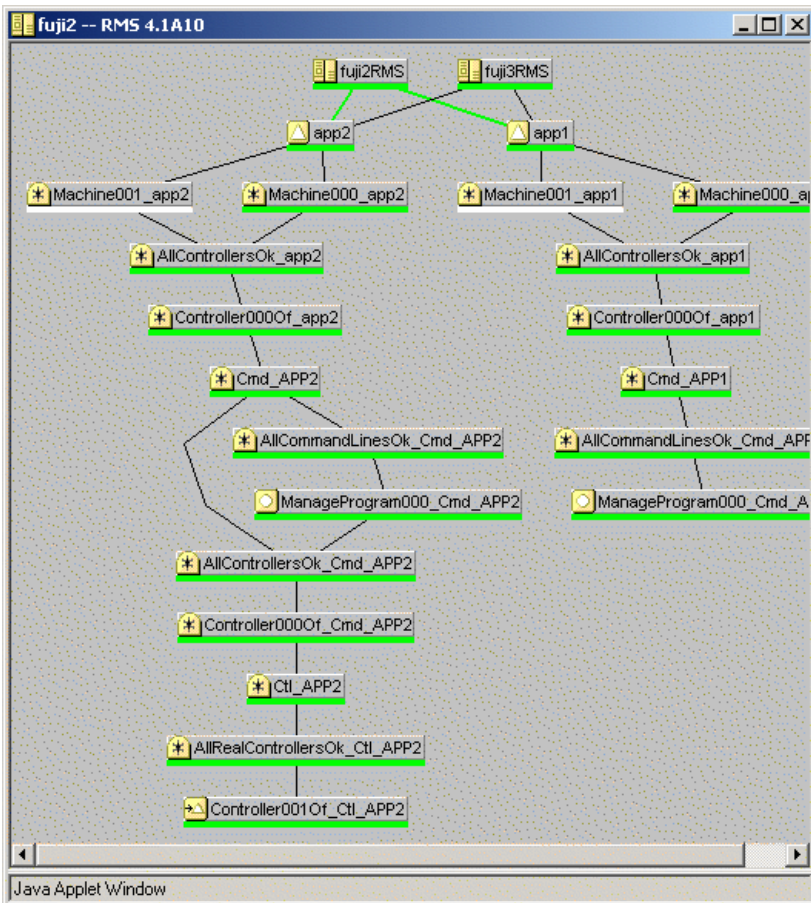


Figure 71: RMS graph customization



#### 4.2.5.8 RMS GUI after it is shut down

After RMS is shut down, the RMS GUI windows become dark gray on the node from which they are getting their information (see Figure 72). In this condition, all the states are white, indicating that the states are unknown. The main window and the clusterwide table continue to show the application states until RMS is shut down on all nodes.

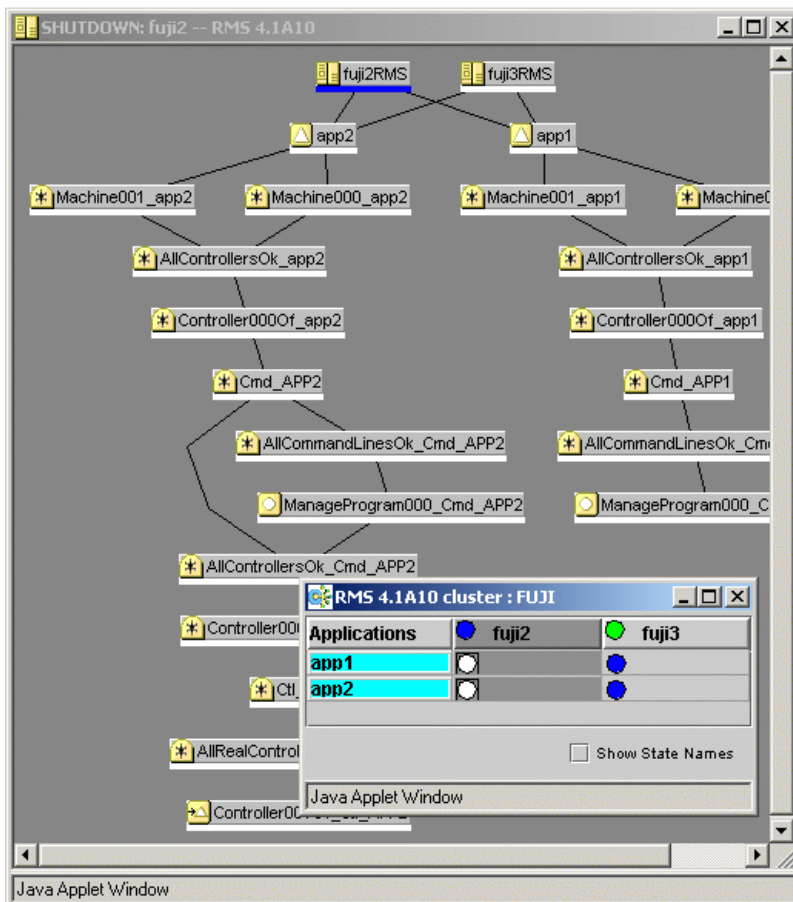


Figure 72: RMS graph after RMS is shut down

## 4.2.6 RMS clusterwide table

The RMS clusterwide table displays the state information about userApplication objects as a summary table. The user can see the state of each of the userApplication objects on each of the system nodes. It presents the information in a concise manner.

Open the clusterwide table through a pop-up menu option for the cluster node (root node) in the RMS tree. The clusterwide table comes up in a separate window (see Figure 73).

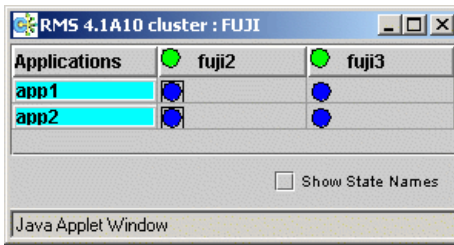


Figure 73: Clusterwide table

You can increase or decrease the size of the clusterwide table window and the size of the columns by using the mouse. If the window is already large enough to fully display all of the table elements, then you will not be allowed to further increase its size.

A square surrounding the colored state circle indicates the primary node for the application. Figure 74 shows that `fuji2` is the primary node for all of the applications.

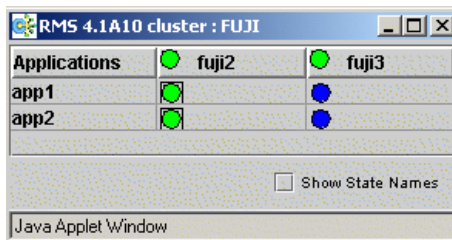


Figure 74: Primary node shown in clusterwide table

Normally, the clusterwide table displays applications in alphabetical order from top to bottom. However, `Faulted` applications are handled specially. If an application is in the `Faulted` state on any node in the cluster, then it is displayed at the top of the table, and the application's name is highlighted by a pink background. This allows the System Administrator to easily spot any `Faulted` applications.

The clusterwide table also makes special provisions for applications that are not `Online` anywhere in the cluster. These applications are also displayed at the top of the table, and the application's name is highlighted in a light blue. Thus, the System Administrator can see what applications are not running anywhere and should probably be brought `Online` on some node.

If there are both `Faulted` applications and applications that are not `Online` anywhere, then the `Faulted` applications are shown above the ones that are not `Online` anywhere.

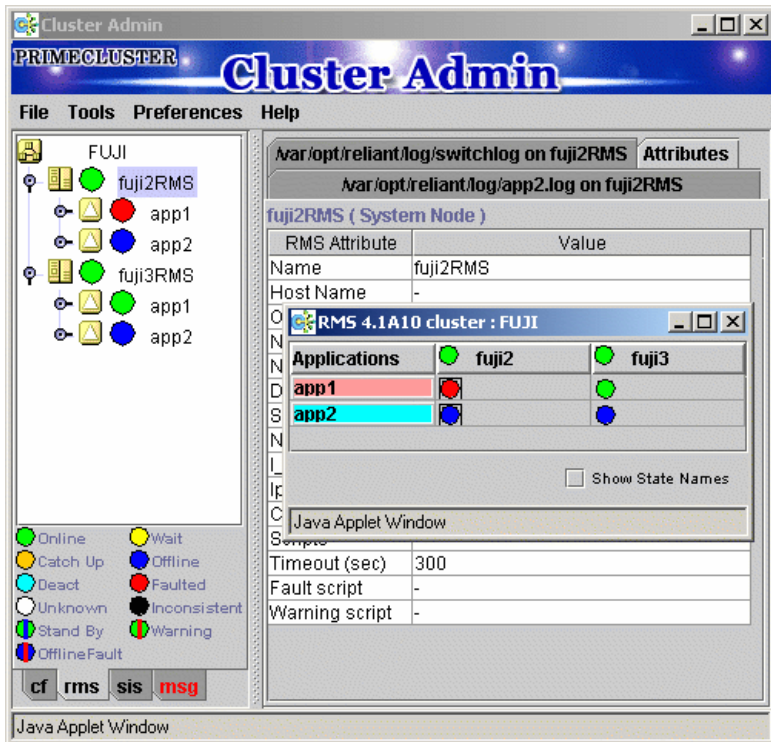


Figure 75: Faulted and Offline applications in the clusterwide table

If there is a split-brain condition in the cluster on both the clusterwide table and the RMS tree, then colored exclamation marks will appear after the colored circles for SysNodes. A colored exclamation mark indicates that the state of that SysNode is different from what another SysNode views it as being. The color of the exclamation mark indicates the state that the other node thinks that the SysNode is in. If there are multiple nodes that see a SysNode in different states, you will see multiple exclamation marks after the colored circle. Exclamation marks are sorted according to the severity of the states. Figure 76 shows a clusterwide table with an application of a split-brain condition.

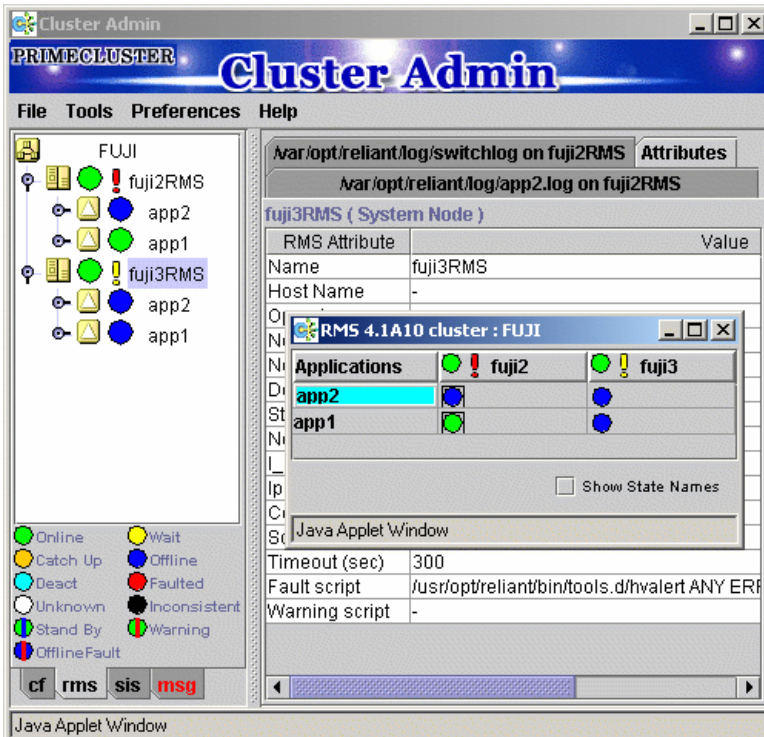


Figure 76: Exclamation marks in clusterwide table and the RMS tree

#### 4.2.6.1 Command pop-ups

Use the context-sensitive command pop-up menus to perform some of the operations on the clusterwide table nodes. Invoke the pop-up menu by right-clicking on an object. The menu options are based on the type and the current state of the selected node (see Figure 77).

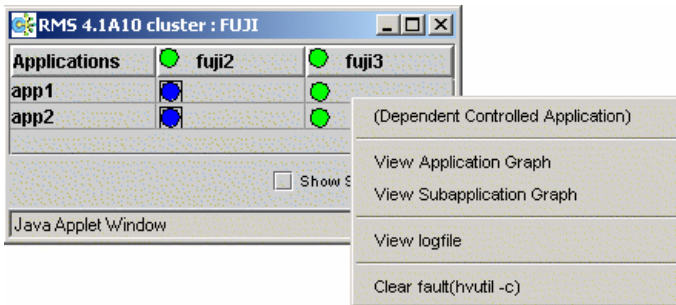


Figure 77: Command pop-ups in clusterwide table

## 4.2.7 Changing the RMS configuration

When you stop and restart RMS with a different configuration, the graphs, the clusterwide table, and the RMS tree are redrawn. In this case, each of the display windows closes and a new display at the same position is displayed.

Figure 78 illustrates the display containing AppA and AppB before RMS is shutdown, and Figure 79 shows the RMS GUI after RMS has been restarted with a different configuration that uses app1 and app2.

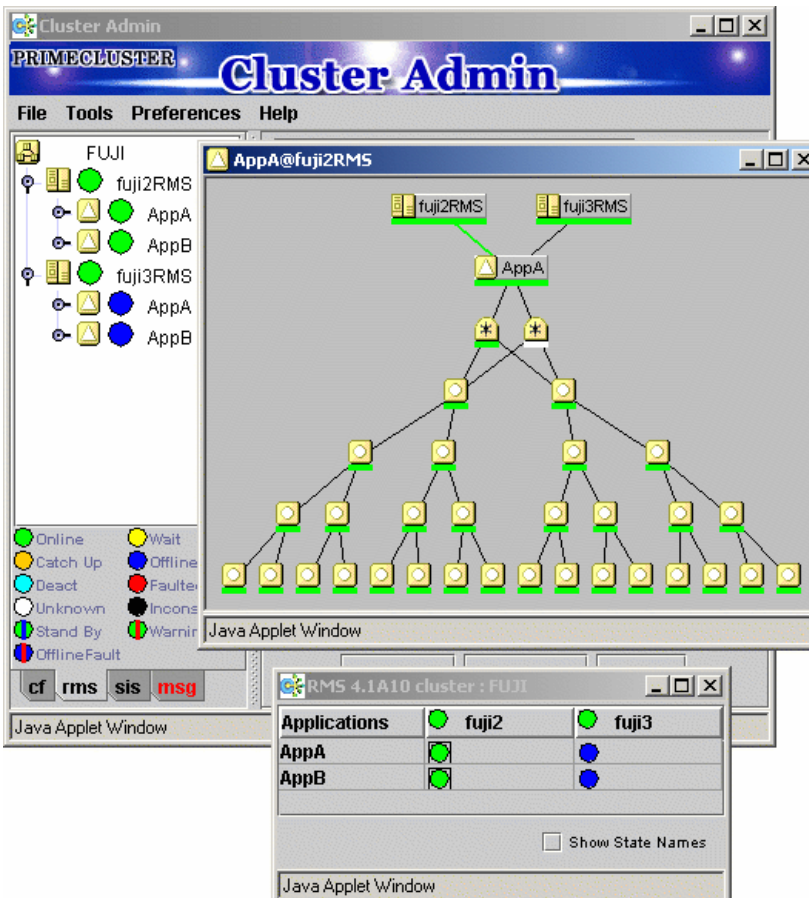


Figure 78: Before RMS is shut down

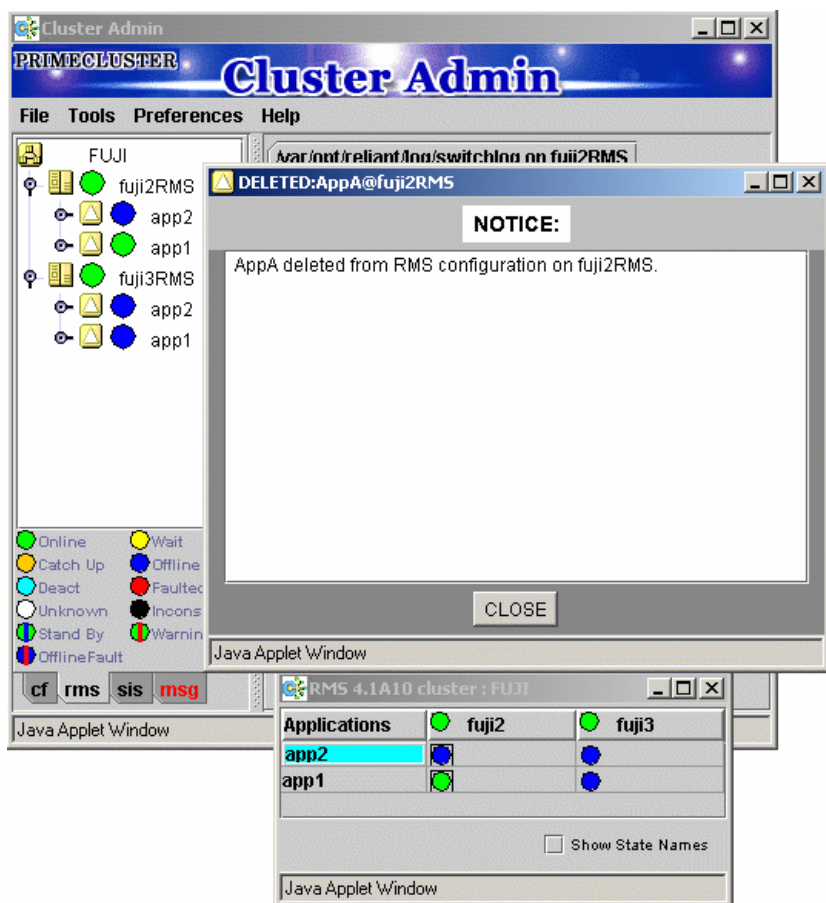


Figure 79: After RMS is restarted with a different configuration

## 4.3 RMS procedures

Using Cluster Admin, you can perform many RMS procedures. These procedures are described briefly in the following sections. These sections are organized as follows:

- GUI options—This is the preferred method of operation.
- CLI options—Refer to the Chapter “Manual pages” for complete details on the CLI options and arguments, some of which are described in this section. For more complete details on CLI options and arguments, refer to the manual page. The commands can also be found in the following directory:

`/opt/SMAW/SMAWRrms/bin`



All the RMS CLI commands take both the CF node names and RMS `SysNode` names for type `SysNode` when the RMS naming convention is followed (that is, names of the form `cfnameRMS`).



### 4.3.1 Starting RMS

RMS can be started on one or all of the nodes in a cluster. You can either use the last start command, or you can enter your own start command. Start RMS from the *RMS Main Toolbar*, located at the top of the Cluster Admin GUI, and select *Start RMS* (see Figure 80).

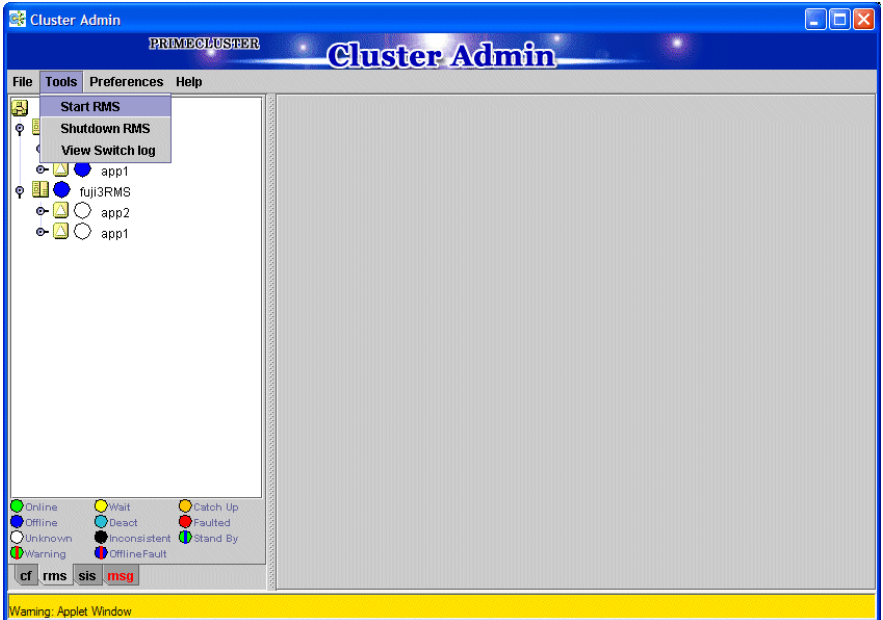


Figure 80: Starting RMS

To start RMS on all nodes, click on *Start RMS* and perform the following steps from the *RMS Start Menu* window:

1. Click on the *Start RMS* radio button for *all available nodes* (see Figure 81).

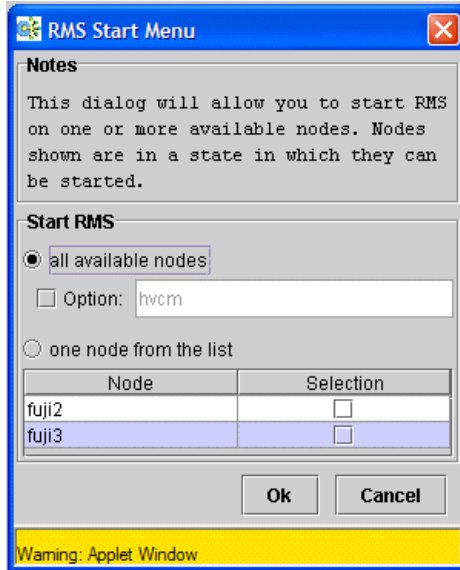


Figure 81: RMS Start Menu for all nodes

2. If you want to specify options to the RMS start command, click in the *Option* box and enter the option in the text field.

To start RMS on individual nodes, perform the following steps from the *RMS Start Menu* window:

1. Click on the *Start RMS* radio button for *one node from the list* and select the desired node or nodes (see Figure 82).

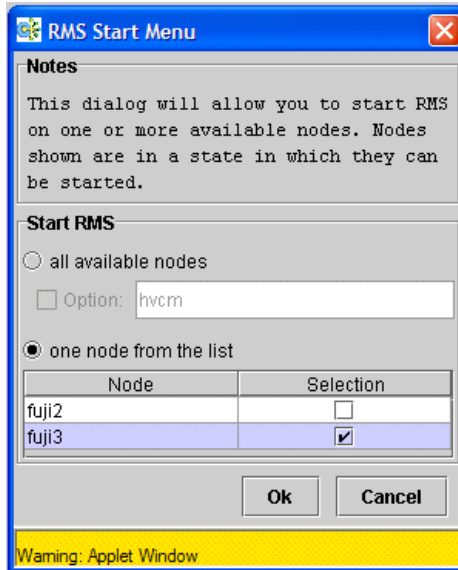


Figure 82: RMS Start Menu for individual nodes

2. If you want to specify options to the RMS start command, click in the *Option* box and enter the option in the text field.

In addition to using the *RMS Main Toolbar*, you can start RMS on individual nodes as follows:

- ▶ Right-click on the system node and select the *StartRMS* option from the pop-up menu to start RMS on that node (see Figure 83).

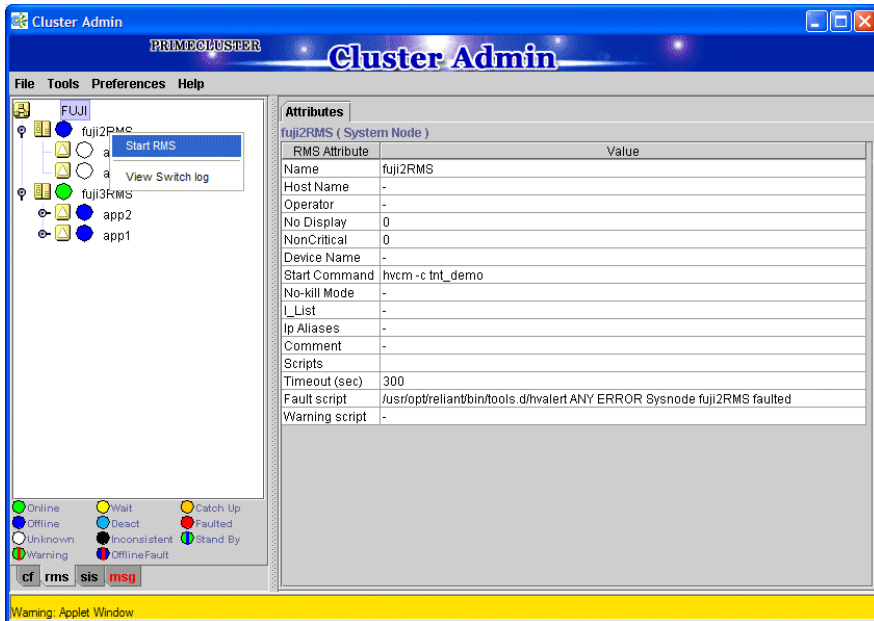


Figure 83: Starting RMS on individual nodes

## CLI

The syntax for the CLI is as follows:

- `hvcn {-a | -s SysNode}`

Starts RMS with the configuration and the options as specified in the default startup file `CONFIG.rms`.

- `hvcn -c config_file {-a | -s SysNode}`

Starts RMS with the specified configuration file.

The `hvcn` command starts the base monitor and the detectors for all monitored resources. In most cases, it is not necessary to specify options to the `hvcn` command; the default values are sufficient for most configurations.

The default startup file, `CONFIG.rms`, can be found in `RELIANT_PATH/etc`. If the default for the environment variable `RELIANT_PATH` has not been changed, RMS searches for `CONFIG.rms` in the default root directory `/opt/SMAW/SMAWRrms/etc`.

- i** The system default run level in `/etc/inittab` must match the chosen RMS start run level; otherwise, the start sequence may be out of order. To verify or to change the RMS run level use the `hvrcllev` command. Refer to the Chapter “Manual pages” for more information.

### 4.3.2 Stopping RMS

Select the *Shutdown RMS* option to stop RMS as follows:

1. Use the *Tools* pull-down menu, or right-click on a system node, and select the mode of shutdown in the subsequent option screen (see Figure 84).

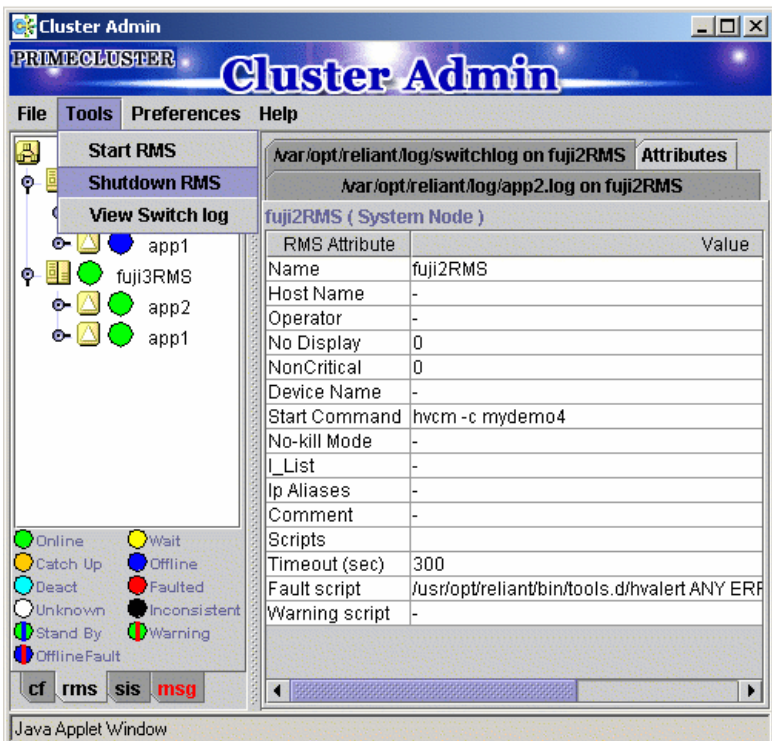


Figure 84: Stopping RMS

2. Select the radio button for *all available nodes* and click *Ok* to shutdown RMS on all nodes (see Figure 85).

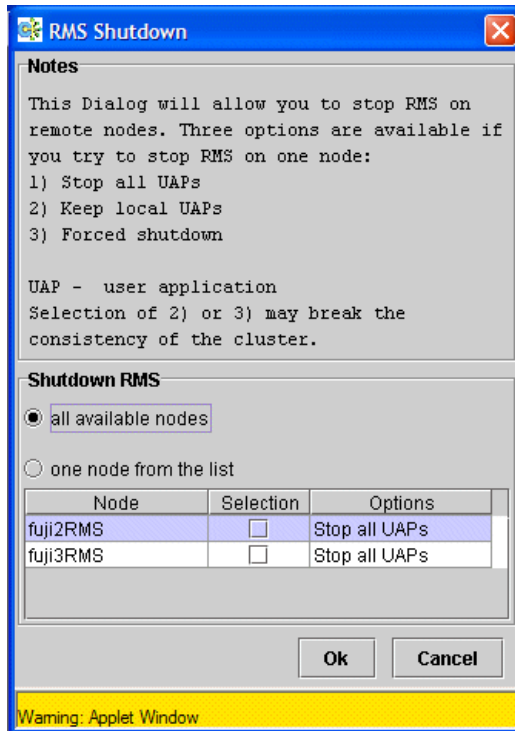


Figure 85: Stopping RMS on all available nodes

3. To shutdown RMS on specific nodes, select the radio button for *one node from the list*, click in the box or boxes of the desired node or nodes to shutdown, and click *Ok* (see Figure 86). This feature gives you the following options under the *Options* column in addition to shutting down RMS on the selected node:
- *Stop all UAPs*—Stops all user applications for the selected node
  - *Keep local UAPs*—Leaves the applications running on the selected node
  - *Forced shutdown*—Performs a forced shutdown of RMS



### Caution

Using a forced shutdown or leaving the applications running and stopping RMS can lead to data inconsistencies or corruption.

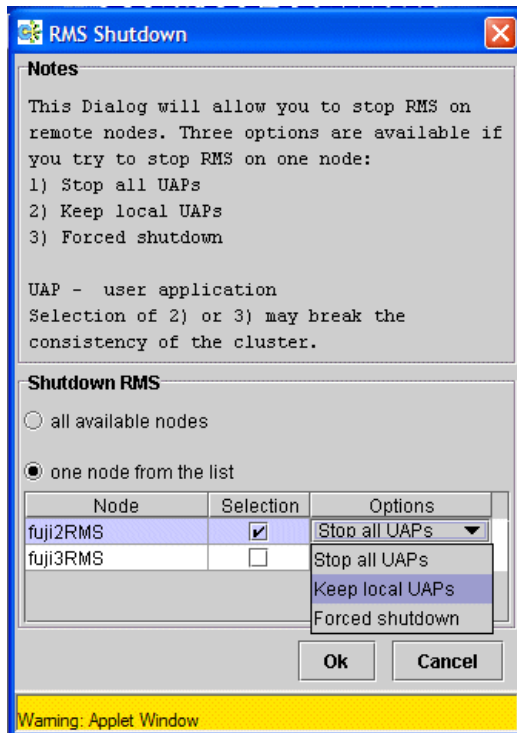


Figure 86: Stopping RMS on one node from the list

Figure 87 shows the command pop-up option to stop RMS on an individual node when you right-click on a system node and select *Shutdown RMS*.

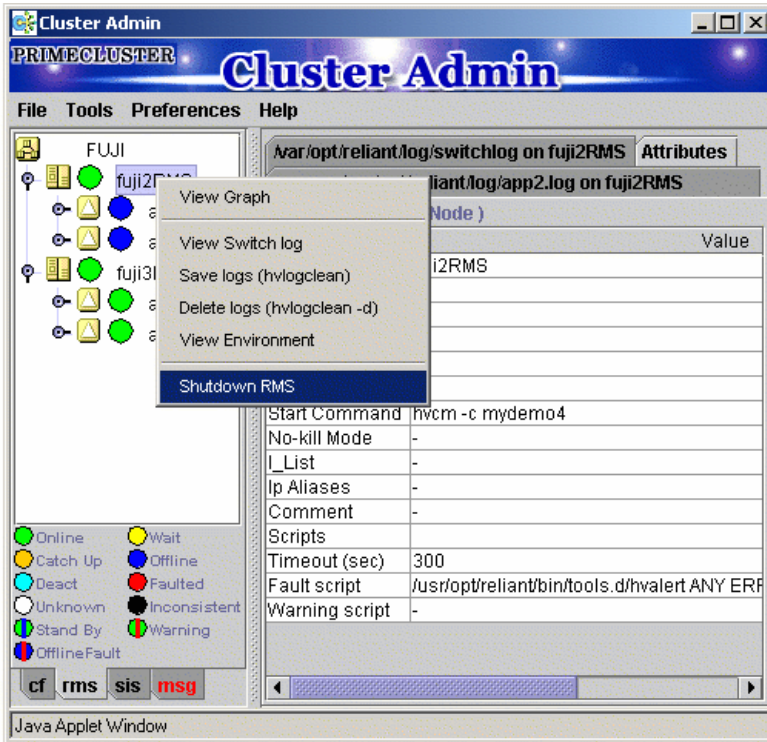


Figure 87: Using command pop-up to stop RMS

## CLI

The syntax for the CLI is as follows:

```
hvshut {-f | -L | -a | -l | -s SysNode }
```

The `hvshut` command shuts down the RMS software on one or more nodes in the configuration. The base monitor on the local node sends a message to other `Online` nodes indicating which node or nodes are to be shut down. The `hvshut` command disables all error detection and recovery routines on the nodes being shut down, but does not shut down the operating system. If any `userApplication` objects are `Online` when the `-f` or `-L` options are used, the applications remain running but are no longer monitored by RMS.





## CLI

The syntax for the CLI is as follows:

```
hvswitch [-f] userApplication [SysNode]
```

The `hvswitch` command manually switches control of a `userApplication` resource from one system node to another in the RMS configuration. The resource being switched must be of type `userApplication`. The system node must be of type `SysNode`. The `-f` option is a forced-switch option.



### Caution

Use the `-f` option carefully as it could result in inconsistencies or data corruption.

### 4.3.4 Switching an application

Switch an Online application as follows:

1. Right-click on the application object and select the *Switch* menu option. A pull-down menu appears listing the available nodes for switchover.
2. Select the target node from the pull-down menu to switch the application to that node (see Figure 89).

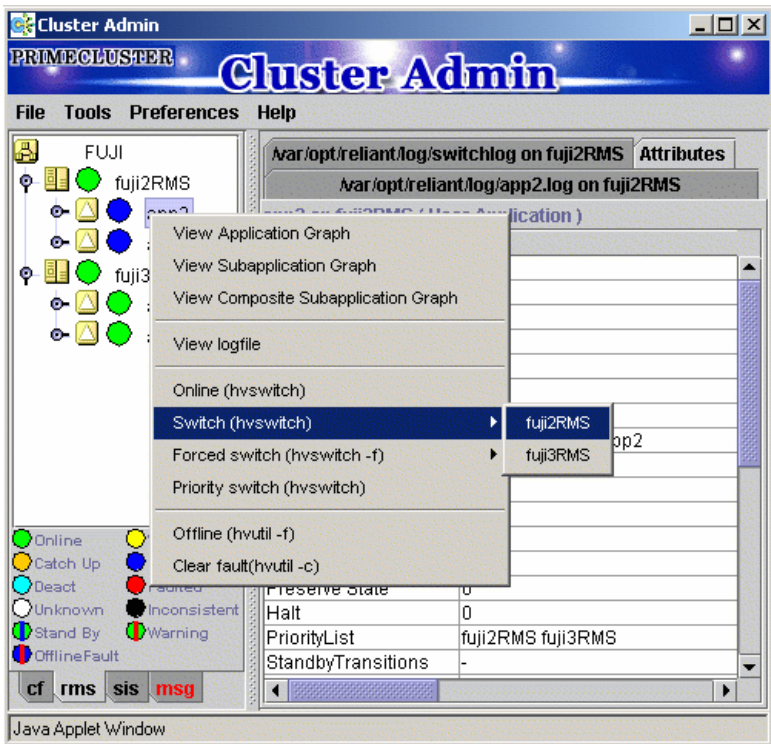


Figure 89: Switching an application



#### Caution

It is recommended that you use the normal mode of switching applications to ensure that application and data consistencies and integrity are maintained. If an application cannot be switched normally, you may use the forced switch mode; however, a forced switch overrides all safety checks and could even result in data corruption or other inconsistencies.

If the application is busy, the command pop-up will not offer the choices to switch the application. Instead, the command pop-up indicates that the application is busy and that you should try later (see Figure 90).

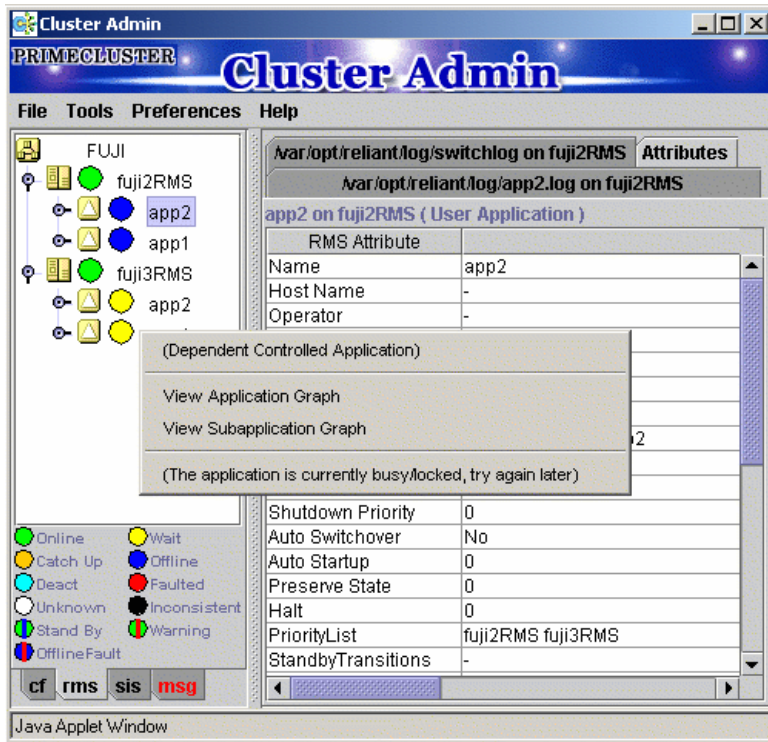


Figure 90: Switching a busy application

## CLI

Please refer to the Section “Starting an application” for information on this command.

### 4.3.5 Activating an application

Activate a deactivated application as follows:

- ▶ Right-click on the application object and select the *Activate* option from the pop-up menu.

Activating an application takes it from the `Deact` state to the `Offline` state. It does not bring it `Online`. Activating a `userApplication` has nothing to do with activating an RMS Wizard configuration. The two concepts are completely independent. You will not need to activate an application unless someone explicitly deactivated that application using the CLI command `hvutil -d userApplication`.

#### CLI

The syntax for the CLI is as follows:

```
hvutil -a userApplication
```



### 4.3.7 Clearing a fault

Clear the fault for an application in the `Faulted` state as follows:

- ▶ Right-click on the application object and select the *Clear Fault* pop-up menu option (see Figure 92).

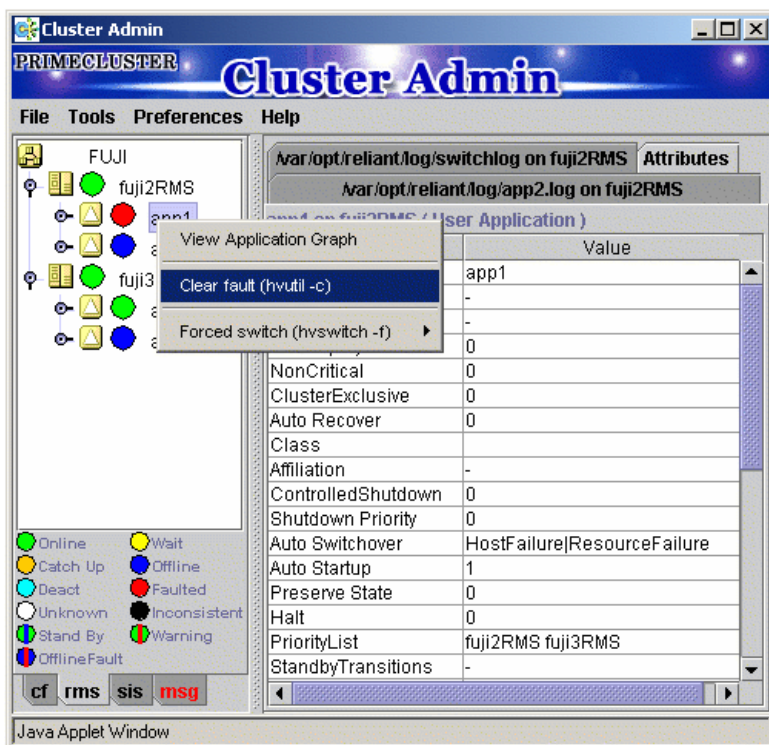


Figure 92: Clearing an application fault

### CLI

The syntax for the CLI is as follows:

```
hvutil -c userApplication
```



If the `userApplication` is in the `Online` state, then clearing the fault will cause RMS to attempt to bring the faulted resource to the `Online` state. If the `userApplication` is in the `Offline` state, then clearing the fault will bring the resource to the `Offline` state.

### 4.3.8 Clearing a SysNode Wait state

Clear any system node in the `Wait` state as follows:

- ▶ Right-click on the system node and select the *Online* or *Offline* option from the pop-up menu.

The clearing of the `Wait` state for a system node will be ignored unless the Shutdown Facility (SF) timeout has been exceeded.



#### Caution

Manually clearing the `SysNode Wait` state by using either `hvutil -o SysNode`, `cftool -k`, or the GUI causes RMS, SF, and CF to believe that the node in question has been confirmed to be down. Doing so without the node really being down can lead to data corruption.

#### CLI

The syntax for the CLI is as follows:

```
hvutil -o SysNode
```

This command clears the `Wait` state for the specified `SysNode` on all cluster nodes after the SF failed to kill the cluster node (`SysNode`) by returning the specified `SysNode` to the `Online` state. If the `SysNode` is currently in the `Wait` state, and if the last detector report for the `SysNode` is in the `Online` state, the `Wait` state is cleared and the `SysNode` goes back to the `Online` state as if no kill request had ever been sent.



### 4.3.9 Displaying environment variables

Display the global or clusterwide environment variables as follows:

- ▶ Right-click on a cluster in the RMS tree window and select *View Environment* (see Figure 93).

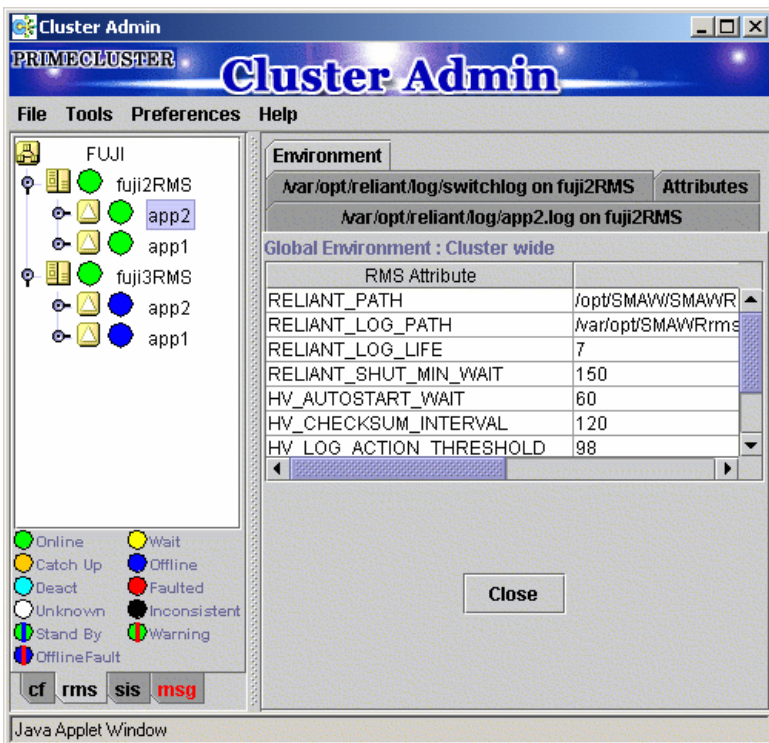


Figure 93: Clusterwide environment variables

Display local environment variables as follows:

- ▶ Right-click on a node in the RMS tree window and select *View Environment* in the command pop-up (see Figure 94).



Figure 94: Local environmental variables pop-up

- i** Displaying the local environment variables displays the clusterwide environment variables as well (see Figure 95).

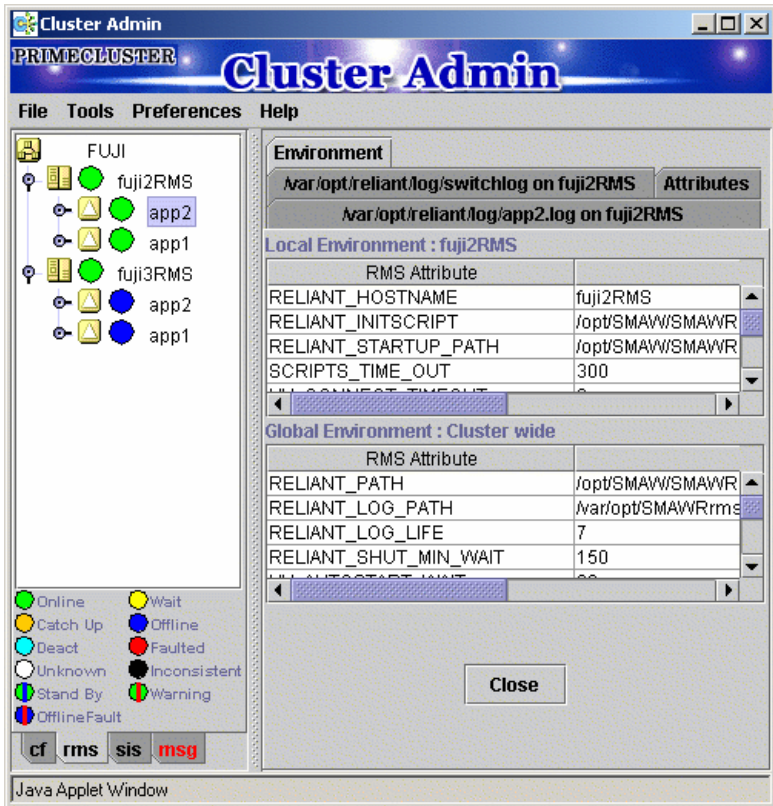


Figure 95: Local environmental variables window

## CLI

The syntax for the CLI is as follows:

```
hvdisp ENV
```

```
hvdisp ENVL
```

### 4.3.10 Displaying application states

The application states of various applications are indicated by different colors. The legend for the application states appears in the RMS main window below the *RMS Tree* panel (see Figure 96).

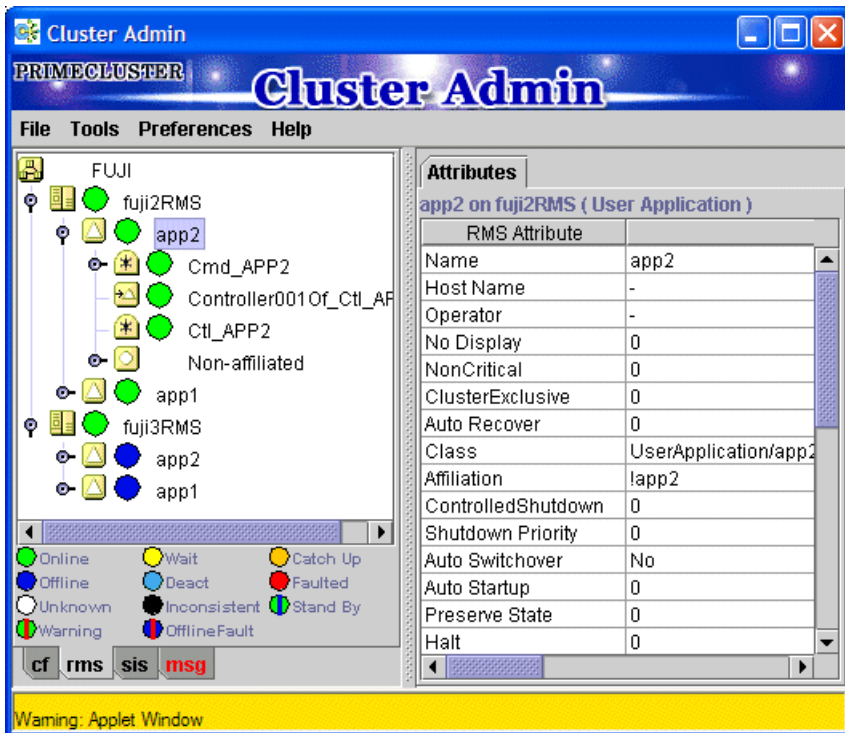


Figure 96: Displaying application states

### CLI

The syntax for the CLI is as follows:

```
hvdisp {-a | -c} [-o out_file]
```

The `-a` option displays the `resource_name`, `resource_type`, `HostName` attribute for each resource in the configuration. The `-c` options displays all information in compact format. The `-o out_file` option is used to send the output to a file called `out_file`. The `hvdisp` command only works when RMS is running.

### 4.3.11 Viewing switchlogs

View the switchlog for a system node as follows:

- ▶ Right-click on the system node and select the *View Switch log* option from the pop-up menu. For more details, refer to the Chapter “Troubleshooting”.

You may search the logs based on keywords, date/time ranges, severity levels, or exit codes using the log viewer.

#### CLI

You can view the switchlog file `/var/opt/SMAWRrms/log/switchlog` using a standard UNIX editor like `vi`. The Chapter “Troubleshooting” describes the RMS log files and their contents.

### 4.3.12 Viewing application logs

View the application logs as follows:

- ▶ Right-click on an application on the RMS tree and choose *View logfile* (for more details, refer to the Chapter “Troubleshooting”).

### 4.3.13 Viewing GUI messages

The *Messages* panel displays error and debug messages related to Cluster Admin. View these messages as follows:

- ▶ Select the *msg* tab on the bottom of the RMS tree panel. This tab turns red if a new message has been added to the text area since it was last viewed.



Message text area can be cleared or detached from the main panel.



---

## 5 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. Finally, the chapter contains a detailed list of all RMS errors and fatal errors with a corresponding description of what the error means and possible actions to correct the problem.

This chapter discusses the following:

- The Section “Overview” summarizes the troubleshooting process.
- The Section “Debug and error messages” describes RMS debug and error messages.
- The Section “Log files” identifies and explains the RMS log files.
- The Section “Using the log viewer” explains the log viewer facilities.
- The Section “Specifying the log level” specifies and explains the log levels.
- The Section “Interpreting log files” explains the meaning of the data in the log files.
- The Section “System log” describes the system log.
- The Section “RMS Wizard logs” details the RMS Wizard log files.
- The Section “RMS error code description” describes the error codes that are used by support engineers and field support for internal debugging.
- The Section “Errors in the switchlog” provides details on the meaning of the error messages that can be found in the switchlog.
- The Section “Fatal error messages” provides details on the meaning of fatal error messages.
- The Section “Additional error messages” provides details on the meaning of the error messages that are printed to the standard out file.
- The Section “RMS troubleshooting” supplies solutions to problems that could occur while using RMS.

## 5.1 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.



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.



Refer to the Section “RMS troubleshooting” 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.

## 5.2 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/SMARrms/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 record in the base monitor 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.



There are also configuration-specific log files in the log directory. It is recommended that administrators evaluate these if necessary. The names of these log files depend on the configuration that was set up

using the RMS Wizards Tools. Consult the RMS Wizards Tools documentation for further information (refer to the Section “Further reading” in the Chapter “Configuration”).

The following log files can also be used for problem solving:

- `hvdet_nodelog`
- `bmlog`

### 5.3 Log files

Table 4 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 it records all modification instructions. The default is off.
base monitor	abortstartlog	This file contains records about bm exit conditions to assist support personnel in determining why RMS failed to start. The following message during startup generates this file: FATAL ERROR: RMS has failed to start!
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” for more information.  Includes all messages received by the base monitor at runtime.  Limited use to administrators since turning on log level flags consumes a great deal of disk space. By default, RMS places no messages in bmlog.

Table 4: Log files

Module	File Name	Contents
Everything (base monitor, generic detector, node detector)	switchlog	Operational events, such as resource switches or bugs. Normally, <i>switchlog</i> is the only log file users need to examine.
generic detector	<i>program</i> log	All messages and job assignments received by the detector. Also contains resource state change information and all error messages. <i>program</i> is the name of the detector under RELIANT_PATH.
node detector (hvdet_node)	hvdet_nodelog	Messages from the built-in node detector, hvdet_node.

Table 4: Log files

## 5.4 Using the log viewer

Invoke the log viewer for the RMS `switchlog` file as follows:

1. Right-click on a SysNode in the RMS Tree.
2. Select *View switch log*.

Figure 97 shows how to invoke the log viewer.

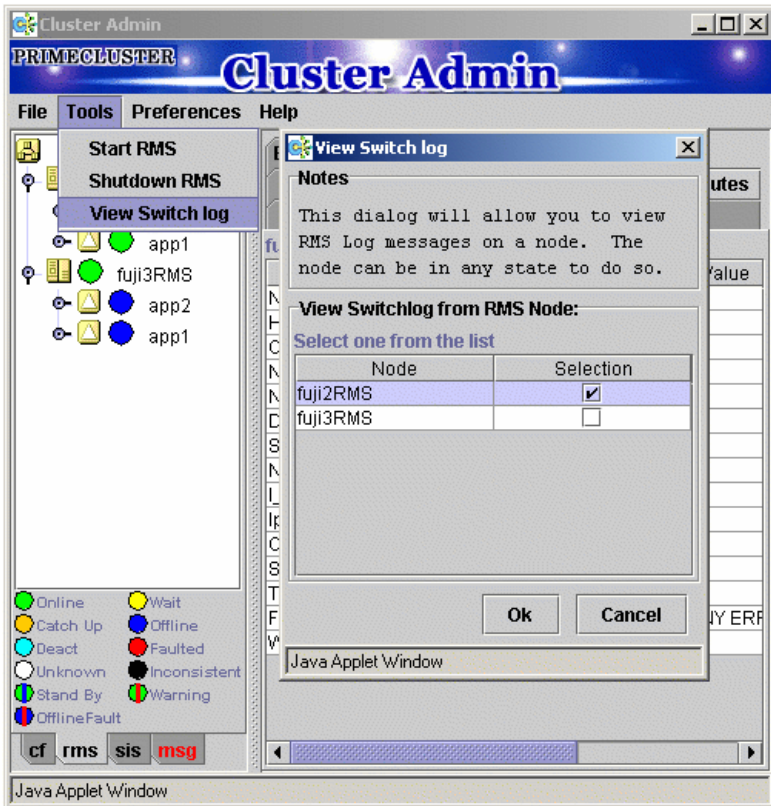


Figure 97: Invoking the log viewer

You can search the logs based on any of the following:

- Resource name
- Date/time range
- Keyword filter
- Severity levels
- Exit codes

You can also search in the log display window by right-clicking on the displayed text. This brings up a *Find* pop-up window (refer to Figure 98).

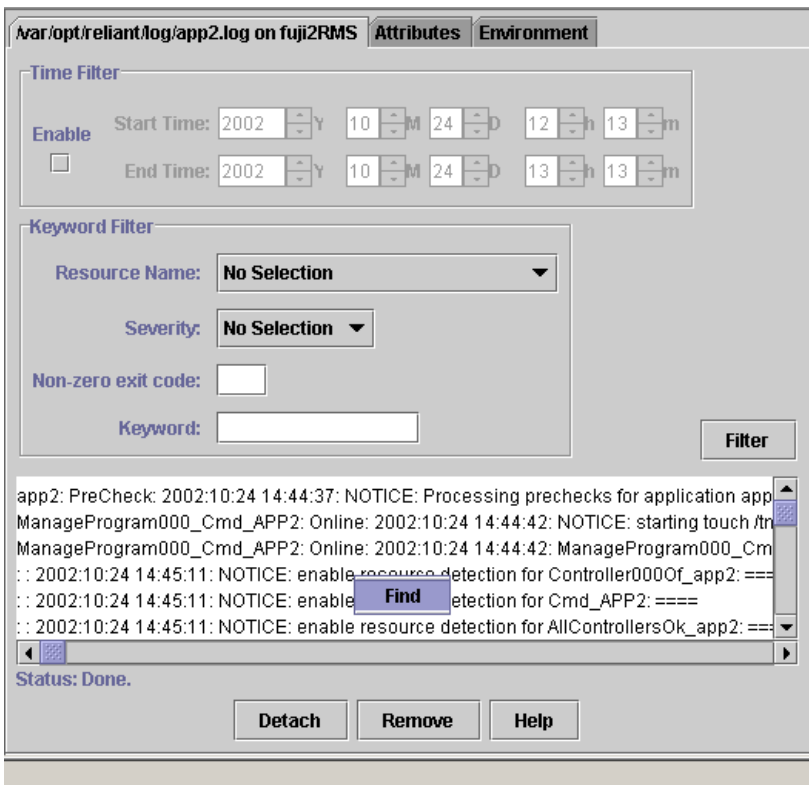


Figure 98: Find pop-up window

Detach the log by clicking on the *Detach* button. Use the *Attach* button to attach it again.

Figure 99 shows a detached log.

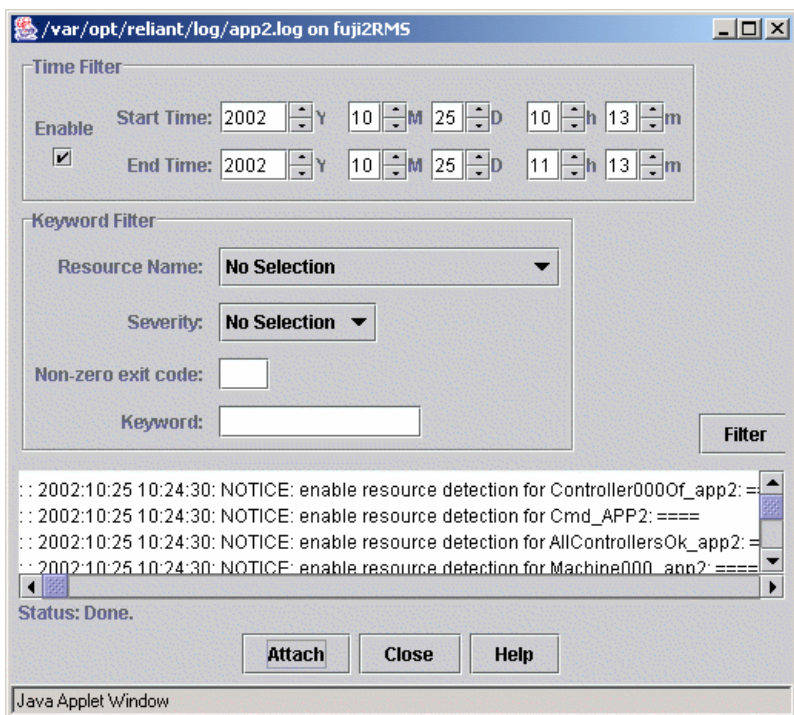


Figure 99: Detached log

## 5.4.1 Search based on resource

Searches based on the name of the resource apply only to application logs. Search the log files based on the name of a resource as follows:

1. Select the name of the resource from the pull-down list.
2. Press the *Filter* button.

Figure 100 shows the window for a search based on the resource name.

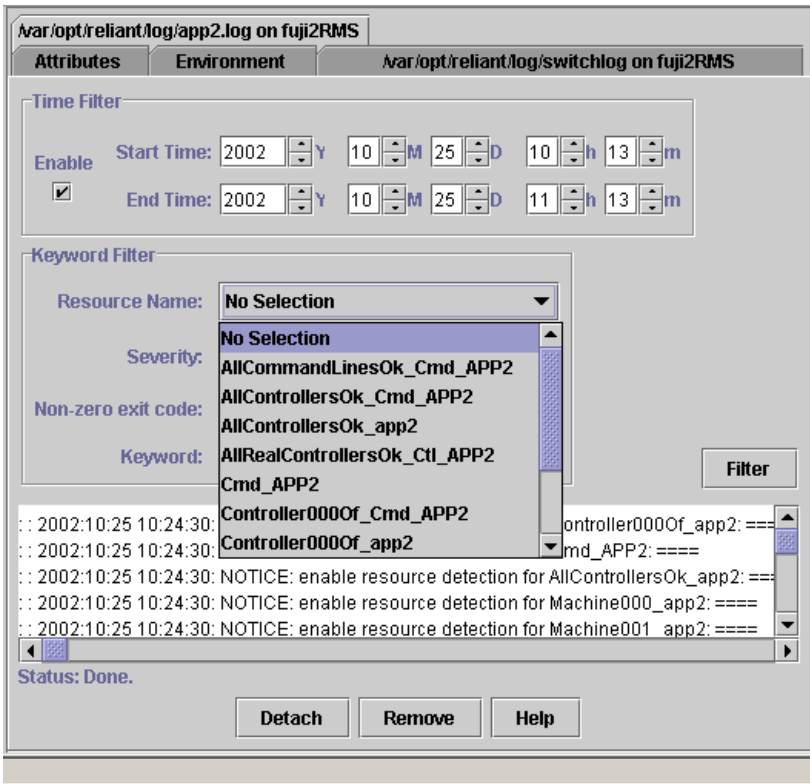


Figure 100: Resource-based search



### 5.4.2 Search based on time

Search the log files based on the date and time range as follows:

1. Specify the start and end times for the search range.
2. Click on *Enable*.
3. Press the *Filter* button.

Figure 101 shows the results for a search based on the time filter.

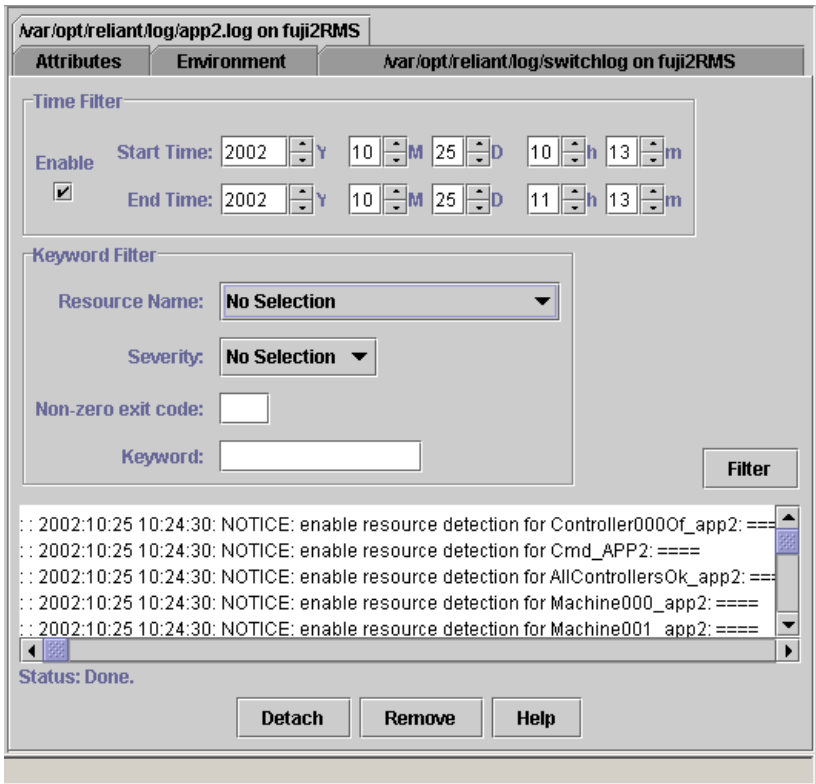


Figure 101: Results of time-based search

### 5.4.3 Search based on keyword

Search the log files based on a keyword as follows:

1. Enter a keyword.
2. Click on the *Filter* button.

Figure 102 shows an example of a log file search based on a keyword.

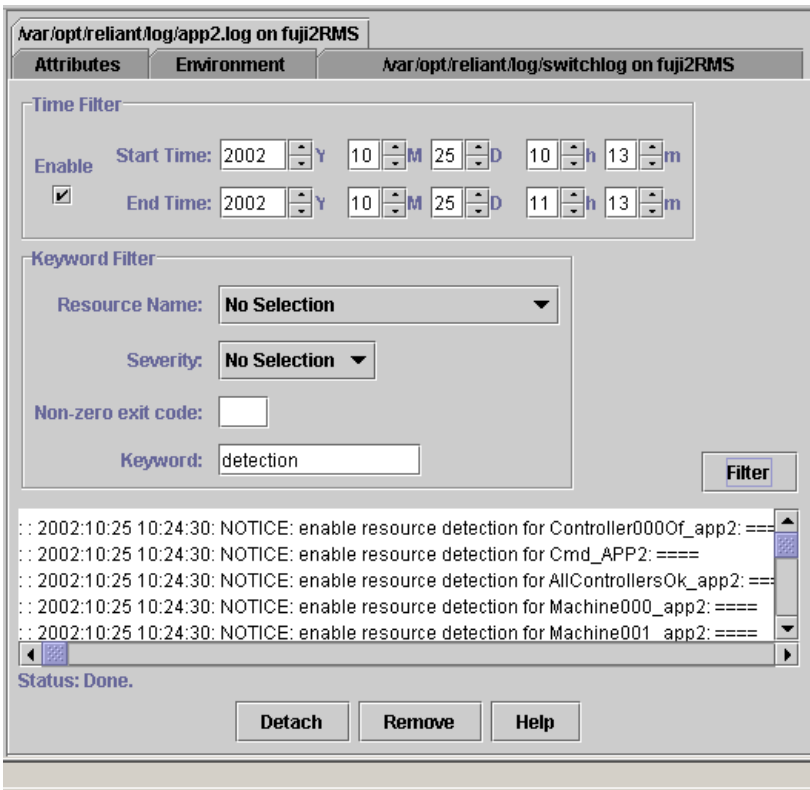


Figure 102: Results of keyword-based search

## 5.4.4 Search based on severity levels

Search the log files based on severity levels as follows:

1. Click on the *Severity* button.
2. Choose one of the severity levels as described in Table 5.
3. Click on the *Filter* button.

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	For information
Debug	Debug messages

Table 5: Descriptions of severity levels

Figure 103 is an example of a log file search based on a severity level.

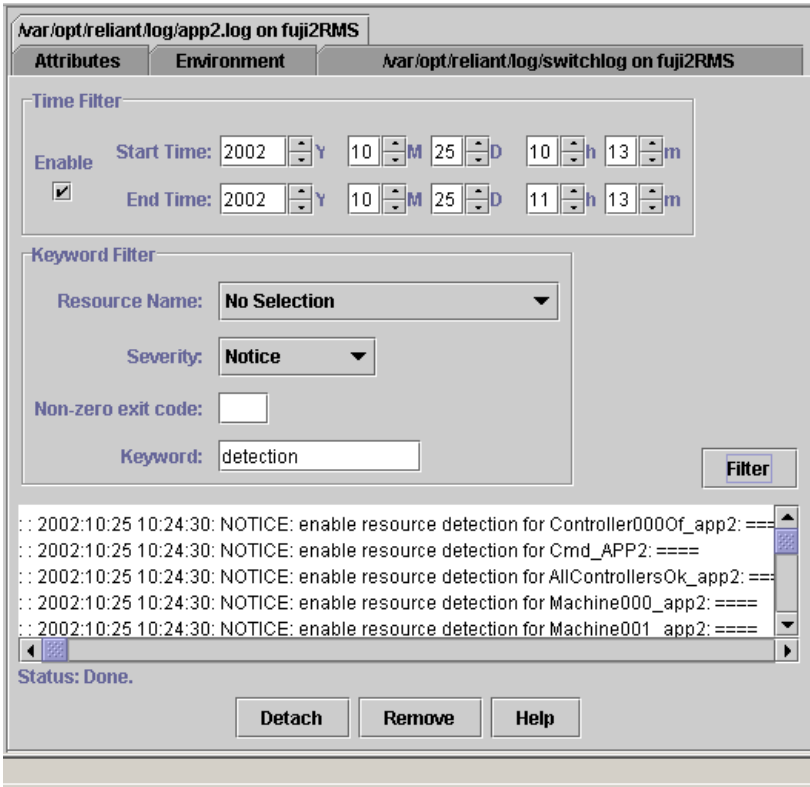



Figure 103: Results of severity-level-based search

## 5.5 Using 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.

## 5.6 Specifying the log level

For further debugging information, use the `-l level` option of the `hvcmm` or `hvtutil` commands to activate various logging procedures.

 Specify logging with the `-l` flag.

To activate logging, shut down RMS and restart it with one of the log levels (refer to Table 6) or use the `hvtutil` 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`. The `-n2` range is the same as `1-n2`. The `n1` range defines 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 defined, all possible log levels are activated. Valid log levels are listed in Table 6.

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

Table 6: Log levels

Log Level	Meaning
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 6: Log levels

## 5.7 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:

<code>switchlog</code>	Records RMS events relevant to the user, such as switch requests and fault indications.
<code>program_name</code> log	Records trace messages or error messages. For example, <code>bmlog</code> . The prefix for trace messages is as follows: <i>time:file:line:</i> . The prefix for error messages is as follows: <i>time:file:line:ERROR</i>

### 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.: =====
```

## 5.8 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`



`hvlogcontrol` is called automatically from the `crontab` file, so there is no manual page.



## 5.9 RMS Wizard logs

The RMS Wizards 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 Wizards 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 Wizards detector logging”.

Unlike RMS, which logs most of its messages in the `switchlog` file, the RMS Wizards 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 Wizard process that is run, 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. The RMS Wizards record resource state 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.gxxlog`—These are detector log files which record all relevant information regarding the resources they are monitoring, like all state transitions.

The format of most RMS Wizard messages is as follows:

```
resource_name:state:timestamp:message_type:Message:delimiter
```

There is a colon-space (:) 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 Wizards 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`, 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` is the number of seconds 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.

## 5.9.1 RMS Wizards detector logging

The RMS Wizard detectors log information to both the `switchlog` file and to their own detector log file `hvdet_XXX.gxxlog` (for example, `hvdet_icmp.g64log`). 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 lesser 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 valid range of values is 1 to 9 (default value is 1). This can be modified in the `hvw` command as follows:

1. Select the *Configuration-Edit-Global-Settings* menu.
2. Choose the *DetectorDetails* sub-menu.
3. Select *MemoryLogLevel*.

When an unexpected `Offline` or `Fault` resource state occurs, the debugging messages are printed from the circular buffer into the detector log file. The information is intended to help determine why the unexpected status report occurred. Because the circular buffer stores earlier logging messages, the log file will contain several `DEBUG` statements with dates prior to the last reported item appearing prior to printing out the circular buffer. The reason for keeping and printing the circular buffer is that a problem has occurred and with the aid of the debugging statements printed from the circular, it can be determined why the detector reported an unexpected resource state change.

## 5.9.2 Modifying levels while RMS is running

It is now possible to turn debug reporting on or off within the RMS Wizard detectors dynamically 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 something greater than zero, debugging is turned on. The greater the value, the more debugging information that is printed; however, the contents of logs may vary from detector to detector. The valid range of values is 1 to 9 (0 means logging is turned off). Any modification to this setting takes affect the next time the configuration is activated.

The command actually creates the file `RELIANT_LOG_PATH/etc/wizardloglevel`, with its contents being the numerical value of the desired debug level. A value of zero in the file turns debugging off.

Alternatively, you can create the file `RELIANT_LOG_PATH/etc/wizardloglevel` manually. If the file exists, a default debugging level of 3 is used. The debug level can be modified by inserting a numerical value in the file.




It is important to realize that by turning on debugging in this manner, all detectors will be affected and print out the additional debugging information.

Be aware that turning on the debugging levels in this manner should only be done when problems occur and for debugging purposes. Once any problems are resolved, debugging should again be turned off so as not to unnecessarily fill up the file system with extraneous information and cause the file system to fill.

## 5.10 RMS error code description

Each error message is preceded by one of the codes listed below. The error codes are related to the following areas:

 The RMS error code description is only used by the support engineers and members of field support for internal debugging.

ADC:	<b>Admin configuration</b>
ADM:	<b>Admin queues, command queues, and detector queues</b>
BM:	<b>Base monitor</b>
CML:	<b>Command line</b>
CRT:	<b>Contracts and contract jobs</b>
CTL:	<b>Controllers</b>
CUP:	<b>userApplication contracts</b>
DET:	<b>Detectors</b>
HVC:	<b>hvcn command</b>
INI:	<b>init script</b>
SCR:	<b>Scripts</b>
SWT:	<b>Switch requests</b>
SYS:	<b>SysNode</b>
UAP:	<b>userApplications</b>
US:	<b>us files</b>
WLT:	<b>Wait list</b>
WRP:	<b>Wrappers</b>

## 5.11 Errors in the switchlog



The RMS product contains a feature for dynamically modifying the RMS configuration file. This feature is used both during RMS startup to establish the running configuration and by the currently unreleased RMS command `hvmod` to perform runtime modification of the running configuration. The error messages for dynamic reconfiguration in this section apply to the base monitor internal use of the feature only.

The following are possible error messages in the `switchlog` file, their potential cause or causes, and the recommended corrective action:

- (ADC, 1): ERROR: Since this host `nodename` has been online for no more than `time` seconds and due to the previous error, it will shutdown now.

This message could appear in the following scenarios:

1. All of the following conditions apply:
  - a) If the checksums of the configurations of the local and the remote node are different when the remote node is joining the cluster
  - b) If no more than `time` seconds have elapsed (`time` is the value of the environment variable `HV_CHECKSUM_INTERVAL` if it is set, or equal to 120 seconds if it is not set)
  - c) If all the `userApplications` on the local node are either `Offline` or `Faulted`
  - d) Then RMS prints the above message and exits with exit code 60
2. All of the following conditions apply:
  - a) If the checksums of the configurations of the local and the remote node are different
  - b) If the configuration for the local node does not include the remote node while the configuration for the remote node does include the local node
  - c) If no more than `time` seconds has elapsed (`time` is the value of the environment variable `HV_CHECKSUM_INTERVAL` if it is set, or equal to 120 seconds if it is not set)
  - d) Then the local node `nodename` shuts down and exits with exit code 60

*Action:*

This occurs because the local and the remote nodes are running different configurations. Ensure that both of them are running the same configuration.

- (ADC, 2): ERROR: Since not all of the applications are offline or faulted on this host <nodename>, 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 could appear if all of the following conditions apply:

1. If the checksums of the configurations of the local and the remote node are different
2. If no more than the number of seconds determined by the value of the environment variable HV\_CHECKSUM\_INTERVAL have passed
3. If not all of the userApplications are Offline or Faulted
4. Then RMS prints the above message and remains Online, and neither automatic nor manual switchover will be possible on this node until the detector *detector* reports Offline or Faulted

*Action:*

Ensure that both the local and the remote node are running the same configuration.

- (ADC, 3): ERROR: Remote host <nodename> reported the checksum (*remote\_checksum*) which is different from the local checksum (*local\_checksum*).

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

*Action:*

The most likely cause for this would be that the local node and the remote node are running configuration files that differ. Check the local and remote nodes as follows:

1. Ensure that the systems are running same configuration; that is, the configuration must be distributed to all systems in the cluster.

2. Verify that the RMS global environment variables are the same on all systems in the cluster.
  3. Ensure that the RMS package installed on all machines has the same version and load.
- (ADC, 4): ERROR: Host *<nodename>* is not in the local configuration.

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

*Action:*

Ensure that the local and the remote node are running the same configuration.

- (ADC, 5): ERROR: Since this host *<nodename>* 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 node are different and if more than *time* seconds have elapsed since this node has gone online (*time* is the value of the environment variable `HV_CHECKSUM_INTERVAL`, if set or equal to 120 seconds if not), then RMS prints the above message.

*Action:*

Ensure that all the nodes in the cluster are running the same configuration file.

- (ADC, 15): ERROR: Global environmental variable *<env\_attribute>* is not set up in *hvenv* file.

This message is the result of RMS being unable to set the global environment variable *<env\_attribute>* because it has not been set in *hvenv*. *env\_attribute* 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`, 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): ERROR: <nodename> 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 state Wait.

- (ADC, 18): ERROR: Local environmental variable <env\_attribute> is not set up in hvenv file.

This message is the result of RMS being unable to set the local environment variable <env\_attribute> because it has not been set in hvenv. env\_attribute 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 the environment variable to an appropriate value.

- (ADC, 20): ERROR: <nodename> is not in the Wait state. hvutil -o request skipped!

When hvutil -o 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 -o was issued prematurely, then reissue the command once the node has reached the state Wait.

- (ADC, 25): ERROR: Application <UserApplication> is locked or busy, modification request skipped.

If hvmod is invoked without the -l option and the userApplication is busy or some requests are being processed, or userApplication contracting is ongoing, then this message appears.

*Action:*

Reissue the `hvmmod` command when the `userApplication` has completed the current switch request.

- (ADC, 27): ERROR: Dynamic modification failed due to inconsistently modified resources.

This message will be printed to the switchlog if the dynamic modification has failed. The exact reason for the failure of dynamic modification is also printed as an error message before this message has been printed.

*Action:*

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

- (ADC, 30): ERROR: HV\_WAIT\_CONFIG value *<seconds>* is incorrect, using 120 instead.

If the value of the environment variable `HV_WAIT_CONFIG` is 0 or a negative number, or has not been set, the default value of 120 is used instead and this message appears in the switchlog.

*Action:*

Set the value of `HV_WAIT_CONFIG` in `/usr/opt/reliant/bin/hvencv`.

- (ADC, 31): ERROR: 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, this message is printed to the switchlog and the operation is aborted. This operation can be any one of the following: `hvrpc`, `hvcopy`.

*Action:*

Contact field support.

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

When error occurs while transferring file *<file>* across the network, this message is the result.

*Action:*

Check if there are any problems with the network.

- (ADC, 33): ERROR: Dynamic modification timeout.

This error message will appear if the time taken for dynamic modification is greater than the timeout value. This timeout is equal to the value of the environment variable `MODIFYTIMEOUTLIMIT` if it is greater than 0 or else it is equal to 0 if the value of the environment variable is less than or equal to 0. If the environment variable itself is not defined then, the timeout value is 120 seconds by default.

*Action:*

Contact field support.

- (ADC, 34): ERROR: Dynamic modification timeout during startup, bm will exit.

If the time taken for dynamic modification during bm startup is greater than the timeout value which is determined from the value of the environment variable `MODIFYTIMEOUTLIMIT`; if it is greater than 0 or equal to 0; if the value of the environment variable is less than or equal to 0 or 120 seconds by default if the environment variable is not defined. RMS then exits with exit code 63.

*Action:*

Contact field support.

- (ADC, 35): ERROR: Dynamic modification timeout, bm will exit.

This should never happen.

*Action:*

Contact field support.

- (ADC, 37): ERROR: 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 userApplication `<userapplication>` is Online this message is the result along with dynamic modification aborting.

*Action:*

Switch the userApplication Offline before making the resource critical.

- (ADC, 38): ERROR: 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 out to the switchlog and dynamic modification is aborted.

*Action:*

Ensure that the userApplication has valid children while performing dynamic modification.

- (ADC, 39): ERROR: The putenv() has failed (*reason\_for\_failure*)

The RMS Wizards use the environment variable `HVMOD_HOST` during dynamic modification. This variable holds the name of the node on which `hvmmod` 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 *reason\_for\_failure*.

*Action:*

Check the reason *reason\_for\_failure* in the switchlog to find out why this operation has failed and take corrective action based on this.

- (ADC, 41): ERROR: The Wizard action failed (*command*)

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

*Action:*

Check the switchlog for the reason for this failure and rectify it before reissuing the `hvmmod` 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 nodes of the cluster. If, for any reason, a file transfer fails, the dynamic modification is aborted.

*Action:*

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

*Action:*

Ensure that node 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 nodes 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:*

Ensure that node and cluster conditions are such that `command` can be safely executed.

- (ADC, 46): ERROR: The file transfer for `<filename>` failed with unequal write byte count, expected `expectedvalue` actual `actual-value`. The dynamic modification will be aborted.

During dynamic modification, files containing modification information are transferred between the nodes 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:*

Ensure that node, cluster, and network conditions are such that *command* can be safely executed.

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

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

*Action:*

Ensure that the file *<filename>* is readable.

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

During a file transfer between the nodes, RMS encountered a problem indicated by the OS error code *errno*.

*Action:*

Ensure that the node, cluster, and network conditions are such that file transfer proceeds without errors.

- (ADC, 49): ERROR: 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:*

Ensure that the node conditions are such that the temporary file can be checked. Sometimes, you may need to restart the *hvdisp* process by hand.

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

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

*Action:*

One of the previous messages contains the detailed explanation about the error occurring while saving the configuration. Correct the node environment according to the explanation, or contact field support.

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

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

*Action:*

Contact field support.

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

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

*Action:*

Contact field support.

- (ADC, 60): ERROR: 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 node, it must be compressed with the command *<command>*. If the *<command>* fails, the operation which requires the file transfer is aborted.

*Action:*

Contact field support.

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

While performing RMS clusterwide shutdown, RMS on node *<node>* failed to shut down.

*Action:*

Contact field support.

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

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

*Action:*

Contact field support.

- (ADC, 63): ERROR: 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: 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): ERROR: Message sequence # is out of sync - File transfer of file *<filename>* has failed.

This should never happen.



*Action:*

Contact field support.

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

During dynamic modification, when you add a new resource or resources to a parent object, and the resources cannot be brought Offline, this message results.

*Action:*

Ensure that the new resource or resources can be brought to the Offline state and reissue the `hvmmod` command.

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

Dynamic modification is aborted if you add a new resource or resources to a parent object that is Online by executing the online scripts, and it cannot be brought Online.

*Action:*

Ensure that the new resource or resources can be brought to the Online state and reissue the `hvmmod` command.

- (ADM, 5): ERROR: 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>* which does not have a parent (and hence not linked to a `userApplication`), this message is printed and dynamic modification is aborted.

*Action:*

Ensure that every object being added during dynamic modification is linked to a `userApplication`.

- (ADM, 6): ERROR: 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 out to the switchlog and dynamic modification aborts.

*Action:*

Ensure that when adding a new resource, its name does not match the name of any other existing resource.

- (ADM, 7): ERROR: 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:*

Ensure that when adding a new resource, its name does not match the name of any other existing resource.

- (ADM, 8): ERROR: 29. Dynamic modification failed: cycle of length *<cycle\_length>* 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 and 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): ERROR: 34. Dynamic modification failed: cannot modify resource *<resource>* since it is going to be deleted.

Deleting a resource causes all of its children with no other parents to be deleted as well; therefore, 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, ensure that the resource that is being modified has not been deleted.

- (ADM, 11): ERROR: 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 node when the parent object has been deleted, the above message appears in the switchlog and dynamic modification is aborted.

*Action:*

Ensure 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): ERROR: 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 appears in the switchlog and dynamic modification is aborted.

*Action:*

Ensure that when a resource is being deleted explicitly, its children have not already been deleted.

- (ADM, 13): ERROR: 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 one of the following 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): ERROR: 48. Dynamic modification failed: cannot link to or unlink from an application <userApplication>.

If the parent *<userApplication>* 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): ERROR: 41. Dynamic modification failed: parent object *<parent\_object>* is not a resource.

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

*Action:*

Ensure that while linking two objects, the parent of the child object is a resource.

- (ADM, 16): ERROR: 42. Dynamic modification failed: child object *<child\_object>* is not a resource.

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

*Action:*

Ensure that while linking two objects, the child of the parent object is a resource.

- (ADM, 17): ERROR: 43. Dynamic modification failed: cannot link parent *<parent\_object>* and child *<child\_object>* since they are already linked.

Trying to link a parent *<parent\_object>* and a child *<child\_object>* which are already linked results in this message. Dynamic modification will be aborted.

*Action:*

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

- (ADM, 18): ERROR: 49. Dynamic modification failed: cannot link a faulted child *<child\_object>* to parent *<parent\_object>* which is not faulted.

While creating a new link between two existing objects during dynamic modification, a faulted child *<child\_object>* cannot be linked to a parent *<parent\_object>* which 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): ERROR: 50. Dynamic modification failed: cannot link child *<child\_object>* which is not online to online parent *<parent\_object>*.

While linking two 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 *<child\_object>* first needs to be brought to the **Online** state before linking it to the online parent *<parent\_object>*.

- (ADM, 20): ERROR: 51. Dynamic modification failed: cannot link child *<child\_object>* which is neither offline nor standby to offline or standby parent *<parent\_object>*.

Any attempt to link two existing objects in which the child is neither in the **Offline** state nor the **Standby** state and the parent is in the **Offline** or **Standby** states is prohibited and results in the message being written to the switchlog. Dynamic modification is aborted.

*Action:*

The child needs to be first brought to the **Offline** or **Standby** state before linking it to the parent which is in the **Offline** or **Standby** state.

- (ADM, 21): ERROR: 44. Dynamic modification failed: Cannot unlink parent *<parent\_object>* and child *<child\_object>* since they are not linked.

Trying to unlink object `<parent_object>` from object `<child_object>` when they are not already linked results in this message with dynamic modification aborted.

*Action:*

If you want to unlink two nodes, ensure that the two objects have a parent child relationship.

- (ADM, 22): ERROR: 46. Dynamic modification failed: child `<child_object>` will be unlinked but not linked back to any of the applications.

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

*Action:*

Ensure that the child is still linked to a `userApplication`.

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

**Description:** Dynamic modification performs some sanity checks. Sanity checks ensure the following:

1. The `HostName` attribute is present only for children of `userApplications`.
2. The child of a `userApplication` does not have another parent.
3. An object belongs to only one `userApplication`.
4. Leaf objects have detectors.
5. Leaf objects which have the `DeviceName` attribute have a valid value for the attribute.
6. The length of the attribute `rName` for the leaf objects is smaller than the maximum.
7. There are no duplicate lines in the `hvgdstartup` file.
8. The `kind` argument for the detector in the `hvgdstartup` is specified.
9. The detector can be loaded.
10. A valid value has been specified for the `rKind` attribute.
11. The `ScriptTimeout` value is greater than the detector cycle time.

12. `ClusterExclusive` and `LieOffline` are mutually exclusive and not used together.

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

*Action:*

Ensure that the sanity checks mentioned above pass.

- (ADM, 24): ERROR: 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 out to the switchlog.

*Action:*

Ensure that the operations of deletion and unlinking are not performed on an object at the same time.

- (ADM, 25): ERROR: 1. Dynamic modification failed: parent object `<parent_object>` is absent.

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

*Action:*

Ensure that the parent specified for a new object that is being added is existent.

- (ADM, 26): ERROR: 18. Dynamic modification failed: parent object `<parent_object>` is neither a resource nor an application.

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

*Action:*

Ensure that the parent object specified for a new object is a resource.

- (ADM, 27): ERROR: 2. Dynamic modification failed -- child object *<child\_object>* is absent.

Any attempt to link to a child object *<child\_object>* which is non-existent leads to this message. Dynamic modification is aborted.

*Action:*

Ensure that the child object specified exists.

- (ADM, 28): ERROR: 19. Dynamic modification failed: child object *<child\_object>* is not a resource.

When a new object *<child\_object>* being added to an existing configuration is not a resource, this message is the result. Dynamic modification is aborted.

*Action:*

Ensure that the child object specified is a resource.

- (ADM, 29): ERROR: 3. Dynamic modification failed -- parent object *<parent\_object>* is absent.

*Action:*

- (ADM, 30): ERROR: 20. Dynamic modification failed: parent object *<parent\_object>* is not a resource.

During dynamic modification, if there is a request to add a new object which is not a resource as a parent object *<parent\_object>*, this message occurs. Dynamic modification is aborted.

*Action:*

Ensure that the object being added as a parent object is a resource.

- (ADM, 31): ERROR: 4. Dynamic modification failed: child object *<child\_object>* is absent.



As part of the dynamic modification, the child object `<child_object>` that has been specified is non-existent, and this message is the result. Dynamic modification is aborted.

*Action:*

Ensure that the child object that has been specified exists.

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

When adding a new object to the RMS resource graph, the child `<child_object>` of this new object was not a resource. Dynamic modification was aborted.

*Action:*

Ensure that when adding a new object, its child is a resource.

- (ADM, 33): ERROR: 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>` which is either non-existent or not a resource, this message is the result. Dynamic modification is aborted.

*Action:*

Ensure that you do not delete an object that does not exist.

- (ADM, 34): ERROR: 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 (`SysNodes`) can be deleted during dynamic modification.

*Action:*

Do not delete this object, or delete another object.

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

This message occurs 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 either Offline or Standby. The reason for this is that a child object needs to be brought to the state of its parent.

*Action:*

Ensure that both the parents of the resource to be added are in the same state before adding it.

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

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

*Action:*

Ensure that the state of the parent resource is one of the states mentioned above.

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

This message occurs when a new object *object* is being added as a child of a userApplication *<userApplication>*, and the value of its HostName attribute is the same as the value of the HostName attribute of an existing child of the userApplication *<userApplication>*. Dynamic modification is aborted.

*Action:*

Ensure that the HostName attribute of an object that is being added to a userApplication is unique.

- (ADM, 40): ERROR: 25. Dynamic modification failed: a new child *<child\_object>* of existing application *<userApplication>* does not have its `HostName` set to a name of any `SysNode`.

This message occurs when a new child object *<child\_object>* is being added to a userApplication *<userApplication>* during dynamic modification, and the `HostName` attribute is missing for this object. Dynamic modification is aborted.

*Action:*

The first-level object under the userApplication must have a `HostName` attribute.

- (ADM, 41): ERROR: 8. Dynamic modification failed: existing child *<child\_object>* is not online, but needs to be linked with *<parent\_object>* which is supposed to be brought online.

This message occurs when both the parent *<parent\_object>* and the child *<child\_object>* have detectors associated with them, and the state of the child is not `Online`, but it needs to be linked to the parent which is supposed to be `Online`. Dynamic modification is aborted.

*Action:*

Ensure that the parent and the child are in a similar state.

- (ADM, 42): ERROR: 9. Dynamic modification failed: existing child *<child\_object>* is online, but needs to be linked with *<parent\_object>* which is supposed to be brought offline.

Trying to link a child *<child\_object>* which is `Online` to a parent object which is supposed to go `Offline` is not allowed. Dynamic modification is aborted.

*Action:*

Ensure that the parent and the child are in a similar state.

- (ADM, 43): ERROR: 10. Dynamic modification failed: linking the same resource *<child\_object>* to different applications *<userApplication\_1>* and *<userApplication\_2>*.

This message occurs when RMS gets a directive to add a new child object *<child\_object>* that has parent and child resources that belong to different userApplications *<userApplication\_1>* and *<userApplication\_2>*. Dynamic modification is aborted.

*Action:*

When adding a new resource, ensure that it does not have parent and children resources that belong to different userApplications.

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

Any attempt to create an object *<object>* which does not have an existing parent leads to this message. Dynamic modification is aborted.

*Action:*

Ensure that the object *<object>* has an existing node as its parent.

- (ADM, 45): ERROR: 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): ERROR: 12. Dynamic modification failed: linking the same resource *<object>* to different applications *<userApplication1>* and *<userApplication2>*.

This message occurs when RMS gets a directive to add a new child object *<object>* by linking it to parent objects belonging to different userApplications *<userApplication1>* and *<userApplication2>*. Dynamic modification is aborted.

*Action:*

When adding a new child resource, ensure that it does not have as its parents resources belonging to different userApplications.

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

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

*Action:*

When adding a new object, ensure that its parent has not already been deleted.

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

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

*Action:*

Do not try to delete an object belonging to an already deleted `userApplication`.

- (ADM, 49): ERROR: 24. Dynamic modification failed: deleted object *<objectName>* belongs to a deleted application.

Any attempt to delete an object *<objectName>* which belongs to a deleted `userApplication` leads to this error because deleting a `userApplication` deletes all its children including *<objectName>*.

*Action:*

Ensure that before an object is deleted, it does not belong to a `userApplication` that is being deleted.

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

This message occurs when RMS gets a directive to delete an object *<object>* that is a descendant of a new object. Dynamic modification is aborted.

*Action:*

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

- (ADM, 51): ERROR: 15. Dynamic modification failed: cannot link to child *<child\_object>* since it will be deleted.

This message occurs when RMS gets a directive to link to a child *<child\_object>* which is going to be deleted. Dynamic modification is aborted.

*Action:*

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

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

This message occurs when 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. Dynamic modification is aborted.

*Action:*

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

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

Trying to modify the attribute of an object *<object>* which is absent leads to this error. Dynamic modification is aborted.

*Action:*

Modify the attributes of an existing object.

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

This message occurs when RMS receives a directive to modify a node *<object>* with attribute *<attribute>* which has an invalid value. Dynamic modification is aborted.

*Action:*

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

- (ADM, 55): ERROR: Cannot create admin queue.

RMS uses UNIX queues internally for Inter Process Communication. Admin queue is one such queue that is used for communication between RMS and other utilities like `hvutil`, `hvmmod`, `hvshut`, `hvswitch`, and `hvdisp`. If this queue cannot be created by RMS, RMS exits with exit code 50.

*Action:*

Restart RMS.

- (ADM, 57): ERROR: `hvdisp - open failed - filename`.

This message occurs if RMS is unable to open the file `/usr/opt/reliant/locks/.rms.<pid>` for writing when `hvdisp` has been invoked with the `-m` option.

*Action:*

Verify that the directory `/usr/opt/reliant/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 by means of the appropriate administrator operation. If none of these problems apply, but the RMS failure still occurs, contact field support.

- (ADM, 58): ERROR: `hvdisp - open failed - file : error_msg`.

For `hvdisp` (except `hvdisp -m`), if unable to open the file `file` (`/usr/opt/reliant/locks/.rms.<pid>`) for writing, it prints out the reason `error_msg`.

*Action:*

Verify that the directory `/usr/opt/reliant/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 by means of the appropriate administrator operation. If none of these problems apply, but the RMS failure still occurs, contact RMS support.

- (ADM, 59): ERROR: `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 `hvmmod`.

*Action:*

Ensure that before a `hvswitch` is performed, `hvmmod` is not ongoing on userApplication *userApplication*.

- (ADM, 60): ERROR: *<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 the result.

*Action:*

Check the manual page for `hvswitch` for usage information.

- (ADM, 62): ERROR: The attribute `<ShutdownScript>` may not be specified for object *<object>*.

The attribute `ShutdownScript` is a hidden attribute within a `SysNode`. Its value is automatically defined by the RMS base monitor—users cannot change it in any way.

*Action:*

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

- (ADM, 63): ERROR: System name *<SysNode>* is unknown.

This message can occur in three scenarios:

1. If the name of the `SysNode` specified in `hvswitch` is not part of the configuration file (`hvswitch [-f] userApplication [SysNode]`)
2. If the name of the `SysNode` specified for `hvshut -s SysNode` is not a valid one; that is, not part of the configuration file
3. If the name of the `SysNode` specified for `hvutil -{o u}` is unknown (hidden options)

*Action:*

Specify the `SysNode` name from within the `.us` file.

- (ADM, 67): ERROR: `_SYSNODE_` Cannot shutdown.

This message could appear if `hvshut -a` has been invoked and not all of the nodes have replied back with an acknowledgement.



*Action:*

Login to the remote hosts. If RMS is still running perform `hvutil -f <userApplication>` to shutdown the userApplications. If this fails, refer to the `switchlog` and `<application>log` for finding the reason for the problem. If all userApplications have been shutdown correctly, shutdown RMS via a forced shutdown `hvshut -f`. Report the problem to RMS support.

- (ADM, 70): ERROR: 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 has terminated.

- (ADM, 75): ERROR: 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 never occur if the configuration has been set up via RC Wizard utilities.

*Action:*

Contact RC Wizard support.

- (ADM, 76): ERROR: 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:*

Ensure that there is no attempt to modify `<attribute>` within object `<object>`.

- (ADM, 77): ERROR: 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): ERROR: 59. Dynamic modification failed: PriorityList <priority\_list> does not include all the hosts where the application <userApplication> may become Online. Ensure that PriorityList contains all hosts from the HostName attribute of the application's children.

**Set PriorityList for user application <userApplication> to include all of 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): ERROR: 60. Dynamic modification failed: PriorityList <%s> includes hosts where the application <userApplication> may never become Online. Ensure PriorityList contains only hosts from the HostName attributes of the application's children.

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

*Action:*

No duplicate host names should be present in the PriorityList.

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

**If a userApplication <userApplication> uses more parent controllers than specified by the attribute MaxControllers <maxcontroller>, this message is the result and dynamic modification aborts.**

*Action:*

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

- (ADM, 82): ERROR: 62. Dynamic modification failed: cannot delete SysNode <SysNode> 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 from a running configuration if this SysNode is not in one of the states: Unknown, Offline, Wait or Faulted.

*Action:*

Shut down RMS on that host and do the deletion.

- (ADM, 83): ERROR: 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:*

Ensure dynamic modification does not contain 'delete SysNode;' where SysNode is the local SysNode.

- (ADM, 84): ERROR: 64. Dynamic modification failed: cannot add SysNode <SysNode> since its name is not valid.

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

*Action:*

Specify a hostname that is resolvable to a network address.

- (ADM, 85): ERROR: 65. Dynamic modification failed: timeout expired, timeout symbol is <timeout\_symbol>.

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

*Action:*

Ensure 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): ERROR: 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:*

Erase 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 attribute `NullDetector` to 1.

- (ADM, 87): ERROR: 67. Dynamic modification failed: only local attributes such as `DetectorStartScript`, `NullDetector`, or `MonitorOnly` can be modified during local modification (`hvmmod -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): ERROR: 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 the attribute `<attribute>` has been modified more than once for object `<object>`.

*Action:*

Modify the attribute only once per object.

- (ADM, 89): ERROR: 69. Dynamic modification failed: cannot rename existing node <SysNode> to <otherSysNode> because either there is no node named <SysNode>, or another node with the name <otherSysNode> already exists, or a new node with that name is being added, or the node 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 you try to rename an existing node <SysNode> to another node <anotherSysNode> but that node name <anotherSysNode> is not a valid name, or is already used by some other host in the cluster, or the node is not a resource, or it is a controlled application.

*Action:*

Choose another valid host name.

- (ADM, 90): ERROR: 70. Dynamic modification failed: cannot change attribute Resource of the controller object <controller-object> 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:*

Ensure deleted applications are not referred to from any controller.

- (ADM, 91): ERROR: 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> by a controller object <controller> but the user application associated with that resource is going to be deleted.

*Action:*

Ensure controller's resource does not refer to a deleted application.

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

This message is the result of starting RMS with the '-a' option but due to some internal error, RMS could not be started on the remote node. This message should never appear.

*Action:*

Try the `-a` option again, or start RMS manually on each node, or contact field support.

- (ADM, 96): ERROR: Remote startup of RMS failed `<startup_command>`. Reason: `error_reason`.

When RMS cannot be started on remote nodes because the command `<startup-command>` failed.

*Action:*

This may occur when some of the nodes are not reachable or the network is down.

- (ADM, 98): ERROR: 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 object 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): ERROR: 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 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.

*Action:*

Ensure that the applications listed in the resource `<newresource>` are written more than once or invalid.

- (ADM, 100): ERROR: 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): ERROR: 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 `SysNodes` 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): ERROR: 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 userApplication `userapplication` exceeds the maximum allowable limit.

*Action:*

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

- (BAS, 2): ERROR: 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): ERROR: No kind specified in `hvgdstartup`.

In the `hvgdstartup`, if the entry for the detector is not of the form `gN -tN -kN` or the part `-kN` is missing, this message is the result, and since RMS is unable to start, it exits with exit code 23.

*Action:*

Modify the entry for the detector so that the kind for the detector is specified.

- (BAS, 2): ERROR: 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): ERROR: No kind specified in `hvgdstartup`.

In the `hvgdstartup`, if the entry for the detector is not of the form `gN -tN -kN` or the part `-kN` is missing, this message is the result and since RMS is unable to start, it exits with exit code 23.

*Action:*

Modify the entry for the detector so that the kind for the detector is specified.

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

During dynamic modification, if there is an attempt to redefine the kind for the `DetectorStartScript`, this message is the result.



*Action:*

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

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

The *message* can be anyone of the following:

1. 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.
2. In `basic.C:parentsCount(...)`
3. The node *<node>* belongs to more than one `userApplication`, *userApplication* and *userApplication*. Nodes must be children of one and only one `userApplication` node.
4. The node *<node>* is a leaf node and this type *<object>* does not have a detector. Leaf nodes must have detectors.
5. The node *node* has an empty `DeviceName` attribute. This node uses a detector and therefore it needs a valid `DeviceName` attribute.
6. The `rName` is *<%s>*, its length *%d* is larger than max length *%d*.
7. The `DuplicateLineInHvgdstartup` is *<%d>*, so `hvgdstartup` file has duplicate line.
8. The `NoKindSpecifiedForGdet` is *<%d>*, so no kind specified in `hvgdstartup`.
9. Failed to load a detector of kind *<%d>*.
10. The node *node* has an invalid `rKind` attribute. Nodes of type `gResource` must have a valid `rKind` attribute.
11. 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 `ScriptTimeout` value for *%s* (currently *%d* seconds) to be greater than the detector cycle time (currently *%d* seconds).

12. Node *<node>* has no detector while all its children's *<MonitorOnly>* attributes are set to 1.
13. The node *node* has both attributes *\\"LieOffline\\"* and *\\"ClusterExclusive\\"* set. These attributes are incompatible; only one of them may be used.
14. The type of object *<object>* cannot be 'or' and 'and' at the same time.
15. Object *<object>* is of type 'and', its state is online, but not all children are online.

*Action:*

Verify above description and change the configuration appropriately.

- (BAS, 14) ERROR: 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.

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

*Action:*

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

- (BAS, 15): ERROR: 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 (leaf objects) is of type *type* which 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: 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.

This condition should never be detected in RMS. If this message appears in switchlog, it indicates a severe problem in the base monitor.

*Action:*

**Contact field support.**

- (BAS, 17): ERROR: 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*.**

*Action:*

**Specify shorter rName.**

- (BAS, 18): ERROR: 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: ERROR IN CONFIGURATION FILE: The NoKind-SpecifiedForGdet is <kind>, so no kind specified in hvgdstartup.

**If the kind has not been specified for the generic detector in the hvgdstartup, this message is the result.**

*Action:*

**Specify the kind for the generic detector in hvgdstartup.**

- (BAS, 24): ERROR: 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: 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).

This message is the result of having the ScriptTimeout value lesser 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: 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 a severe corruption of the RMS executable.

*Action:*

Contact field support.

- (BAS, 27): ERROR: 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: ERROR IN CONFIGURATION FILE: object *<object>* cannot have its HostName attribute set since it is not a child of any userApplication.

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

*Action:*

Eliminate `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: 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: 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: 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: 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*.

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

When RMS encounters problems in transmitting the message *<message>* to another node in the cluster, it prints this message. This could be because RMS on the other node is down or there might be a network problem.

*Action:*

Ensure that the RMS is running on the other nodes in the cluster and check for network issues.

- (BM, 14): ERROR: S6: local queue is empty on read directive in line: *line\_number*.

RMS internal error.

*Action:*

Contact field support.

- (BM, 15): ERROR: S2: destination object *<object>* is absent in line: *line\_number*.

RMS internal error.

*Action:*

Contact field support.

- (BM, 16): ERROR: S2: sender object *<object>* is absent in line: *line\_number*.

RMS internal error.

*Action:***Contact field support.**

- (BM, 17): ERROR: 53. Dynamic modification failed: line *line\_number*, 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): ERROR: 54. Dynamic modification failed: line *line\_number*, 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): ERROR: 39. Dynamic modification failed: line *line\_number*, cannot modify attribute *<attribute>* of object *<object>* with value *<value>*.  
Invalid attribute is specified for modification.

*Action:***Modify only valid attributes.**

- (BM, 20): ERROR: 77. Dynamic modification failed: line *line\_number*, 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): ERROR: 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): ERROR: 80. Dynamic modification failed: The *<Follow>* attribute for controller *<controller>* is set to 1, but the content of a PriorityList of the controlled application *<controlled\_userApplication>* 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 *<controlled\_userApplication>* is different from the content of the PriorityList of the application *<userApplication>* to which the controller *<controller>* belongs.

*Action:*

Ensure that the PriorityList of the controller and the controlled application is same.

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

During dynamic modification, when a new resource or resources are to be added to a resource that is Standby cannot be brought Standby, this message is the result.

*Action:*

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

- (BM, 25): ERROR: 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:*

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

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

This message appears when the user sets both controller attributes StandbyCapable and IgnoreStandbyRequest to 0.

*Action:*

Ensure that only one is set to 1 and other to 0.

- (BM, 29): ERROR: 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 object *<controller>* should have one of its attributes OnlineTimeout or StandbyTimeout be null to allow the attribute Follow to be 1.

*Action:*

Set the attributes accordingly and try again.

- (BM, 42): ERROR: 87. Dynamic modification failed: application *<userApplication>* is not controlled by any controller, but has one of its attributes ControlledSwitch or ControlledShutdown 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): ERROR: 89. Dynamic modification failed: cannot modify a global attribute *<attribute>* locally on host *<hostname>*.

The user cannot modify global attributes *<attribute>* like DetectorStartScript or NullDetector or MonitorOnly locally on a host *<hostname>*

*Action:*

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

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

CIP configuration file has missing entries.

*Action:*

Ensure that the CIP configuration has entries for all the RMS nodes that are running in a cluster.

- (BM, 59): ERROR: 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.

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

While obtaining message queue parameters, sysdef was unable to communicate them back to the base monitor. The values of *errno* and *reason* indicate the type of error.

*Action:*

Contact field support.

- (BM, 71): ERROR: 90. 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", ControlledSwitch = 1, and ControlledShutdown = 1. However, the real values are IndependentSwitch = *<ISW>*, AutoSwitchOver = *<ASW>*, ControlledSwitch = *<CSW>*, and ControlledShutdown = *<CSS>*.

When the controller's attribute `Follow` is set, other attributes such as `IndependentSwitchover`, `AutoSwitchOver`, `ControlledSwitch`, and `ControlledShutdown` must have the values 0, No, 1, and 1 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): ERROR: 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 application where the controller belongs to.

*Action:*

Check and correct the configuration.

- (BM, 73): ERROR: 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): ERROR: 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): ERROR: 88. Dynamic modification failed: controller `<controller>` has its attributes `SplitRequest`, `IgnoreOnlineRequest`, 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): ERROR: 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): ERROR: 93. Dynamic modification failed: local controller attributes such as `NullDetector` or `MonitorOnly` cannot be modified during local modification (`hvmmod -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): ERROR: 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*.

- (CML, 11): ERROR: Option (*option*) requires an operand.

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

*Action:*

Check the *hvcn* manual page for correct usage.

- (CML, 12): ERROR: Unrecognized option *option*.

The option provided is not a valid one.

*Action:*

Check the *hvcn* manual page for correct usage.

- (CML, 17): ERROR: Incorrect range argument with *-l* option.

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

*Action:*

Check the manual page for *hvcn* for range argument with *-l* option.

- (CML, 18): ERROR: 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 *hvcn* is greater than the maximum possible loglevel *maxloglevel*, this message is the result and RMS exits with exit code 4.

*Action:*

Specify a loglevel between 1 and *maxloglevel* for hvcm -l.

- (CML, 19): ERROR: 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): ERROR: Log level must be numeric.

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

*Action:*

Specify a numeric value for the log level.

- (CML, 21): ERROR: 0 is an invalid range value. 0 implies all values\nIf 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 (0-*somevalue*), this message is printed and RMS exits with exit code 4.

*Action:*

The valid range for the -l option of hvcm is 1..*maxloglevel*.

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

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

*Action:*

Contact field support.

- (CRT, 2): ERROR: cannot obtain the NET\_SEND\_Q queue.

RMS uses internal queues for sending contracts (contracts are messages that are transmitted between the nodes in a cluster and which ensure that the different nodes are synchronized with respect to a particular operation) between processes or on different nodes. If there is a problem with the queue `NET_SEND_Q` which is being used to transmit these contracts from one node to the other in the RMS cluster, this message is printed to the switchlog.

*Action:*

Contact field support.

- (CRT, 3): ERROR: Message send failed.

When RMS tries to send a message to another node in the cluster, if the delivery of this message over the queue `NET_SEND_Q` has failed, this message is the result. This could be due to the fact that the node which 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 nodes in the cluster are all alive and make sure that none of them are experiencing any network problems.

- (CRT, 4): ERROR: Contract retransmit failed: Message Id = `id\n\tsee bmlog for contract details.`

When RMS on one node sends a contract to another node or itself (if there is only one node in the cluster) over the queue `NET_SEND_Q`, it tries to transmit this contract a certain number of times which is determined internally. If this message transmission fails even after all these attempts, this message is printed to the switchlog and this contract is discarded (`userApplication contract is not discarded`).

*Action:*

Ensure that there is no problem with the cluster interconnect. If contract retransmissions occur for `userApplication` contracts, ensure that the cluster is in consistent condition (that is, no `userApplication` is `Online` on more than one node, no `SysNode` is in a `pending Wait` state, and so forth).

- (CRT, 5): ERROR: The contract `<crtnode>` is being dropped because the local host `<crtnode>` has found the host originator `<othernode>` 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 node `crtnode` sees the contract node originator in state `state` when it is expected to be in state Online.

*Action:*

Ensure that the internode communication channels are working correctly and the nodes see each other as Online.

- (CTL, 1): ERROR: Controller `<controller>` will not operate properly since its controlled resource `<resource>` is not in the configuration.

This message appears when a resource is not in the configuration which 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): ERROR: 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:*

Ensure that more than one controlled application for a Controller is not Online.

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



If the cluster nodes are unable to reach an agreement as to which node is responsible for a particular `userApplication`. The most likely reason for this is that due to an erroneous system administrator intervention (for example, a forced `hvs` switch request) the `userApplication` is `Online` on more than one node simultaneously.

*Action:*

Analyze the cluster inconsistency and perform the appropriate action to resolve it. That is, in the case of being `Online` on more than one node: Shutdown (`hvutil -f`) the `userApplication` on all but one node.

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

This message should never appear.

*Action:*

Contact field support.

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

The contract received by the object from the application is not recognizable. This message should never appear.

*Action:*

Contact field support.

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

The application *userApplication* is `Online` on the local node, and it is also `Online` on another node.

*Action:*

User application can only be `Online` on one node. Ensure the application is `Offline` on all but one of the nodes. If this is not the case use `hvutil -f` to bring the `userApplication` to an `Offline` state on the superfluous nodes.

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

If the cluster nodes are unable to reach an agreement as to which node is responsible for a particular `userApplication`. The most likely reason for this is that due to an erroneous system administrator intervention (for example, a forced `hvswitch` request) the `userApplication` is `Online` on more than one node simultaneously.



This message corresponds to (CUP, 2). While (CUP, 8) is printed on the contract originator, (CUP, 2) is printed on the non-originator nodes.

*Action:*

Analyze the cluster inconsistency and perform the appropriate action to resolve it. That is, in the case of being `Online` on more than one node: Shutdown (`hvutil -f`) the `userApplication` on all but one node.

- (DET, 1): ERROR: FAULT REASON: Resource `<resource_name>` transitioned to a `Faulted` state due to a child fault.

This message appears when the child faulted unexpectedly which causes the resource to fault.

*Action:*

No action required.

- (DET, 2): ERROR: 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): ERROR: FAULT REASON: Resource `<resource_name>` 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:*

No action required.

- (DET, 4): ERROR: FAULT REASON: Resource *<resource\_name>* 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 also failed to execute for that resource.

*Action:*

No action required.

- (DET, 5): ERROR: FAULT REASON: Resource *<resource\_name>* 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 to be Faulted.

*Action:*

Ensure that the Offline script transitions the resource into the Offline state.

- (DET, 6): ERROR: FAULT REASON: Resource *<resource\_name>* 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 to be Faulted.

*Action:*

Ensure the Online script moves the resource into Online state.

- (DET, 7): ERROR: FAULT REASON: Resource *<resource\_name>* transitioned to a Faulted state due to the resource unexpectedly becoming Offline.

This message appears when the resource becomes Offline unexpectedly.

*Action:*

No action required.

- (DET, 11): ERROR: DETECTOR STARTUP FAILED: Corrupted command line *<commandline>*.

This message occurs when the command line is empty or has some incorrect value.

This should never happen.

- (DET, 12): ERROR: DETECTOR STARTUP FAILED *<detector>*. REASON: *errorreason*.

If the detector *detector* could not be started due to *errorreason*, this message is the result. 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:*

Take appropriate action, depending on the error reason.

- (DET, 13): ERROR: Failed to execute script *<script>*.

The detector script is no good or the format is no good.

*Action:*

Check the detector startup script.

- (DET, 24): ERROR: FAULT REASON: Resource *<resource\_name>* 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 to be Faulted.

*Action:*

Ensure the Online script moves the resource into Standby or Online state during standby request.

- (DET, 26): ERROR: 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 its online scripts which may transition the state of the resource to faulted.

*Action:*

Ensure that the resource can come Online after executing the Online scripts.

- (DET, 28): ERROR: *<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, which is not defined for the local node. This must never happen.

*Action:*

Contact field support.

- (DET, 34): ERROR "No heartbeat has been received from the detector with pid *<process\_ID>*, *<startup\_command>*, 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, it 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.

*Action:*

Ensure the detector is active using system tools such as `truss(1)`.

- (GEN, 1): ERROR: Usage: *command* -t time\_interval -k kind [-d]

Usage error for *command*.

*Action:*

Contact field support.

- (GEN, 3): ERROR: Cannot open *command* log file.

The file *command* log used for logging could not be opened.

*Action:*

Contact field support.

- (GEN, 4): ERROR: fail to create mutex: *directory*

The various RMS commands like *hvdisp*, *hvswitch*, *hvutil*, *hveject*, *hvjoin*, and *hvdump* utilize the lock files from the *directory* *directory* for signal handling purposes. These files are deleted after these commands are completed. The lock files *directory* *directory* is also cleaned when RMS starts up. If it is not cleaned for some reason, this message is the result. RMS exits with exit code 99.

*Action:*

Ensure that the lock files *directory* *directory* exists.

- (GEN, 5): ERROR: *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.

- (INI, 1): ERROR: 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): ERROR: 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*.

- (MIS, 1): ERROR: No space for object.

*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.

Ensure that the user has root privileges before issuing the command.

- (NOD, 6): ERROR: Usage: *detector* -t *time\_interval*.

If the detector *<detector>* has been provided a non-integer argument, this message is the result and the detector exits with exit code 103.

*Action:*

Provide an integer as the *time\_interval* for the detector *detector*.

- (NOD, 7): ERROR: 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.

- (NOD, 8): ERROR: Usage: *detector* -t *time\_interval* [-d] [-n]

If the argument *-t time\_interval* has not been provided for the detector *<detector>* or if an argument apart from *-d* or *-n* is used, this message is printed to the switchlog and the detector exits with exit code 103.

*Action:*

Follow the right usage for the invocation of the detector.

- (NOD, 9): ERROR: *detector*: Failed to open req\_queue.

The detector `hvdet_node` utilizes the queue `req_q` for getting jobs from the base monitor. If there is some problem with the queue, this message is the result with the detector exiting with exit code 106.

*Action:*

Contact field support.

- (NOD, 10): ERROR: *detector*: Failed to open rep\_queue.

The detector `hvdet_node` utilizes the queue `rep_q` to report the state of the other `SysNodes` in the cluster to the base monitor running on the same host as the detector. If there is a problem in sending the state over to the base monitor, this message is printed out and the detector exits with exit code 112.

*Action:*

Contact field support.

- (NOD, 11): ERROR: *service*: `getservbyname` returned NULL.

If the detector has been unable to find the port at which the service `<service>` resides, this message is printed to the `switchlog` and the detector exits with exit code 126.

*Action:*

This is probably due to the absence of an entry for service `<service>` in `/etc/services`.

- (NOD, 12): ERROR: *detector*: no `NODE_SYS_Q`.

The detector `hvdet_node` uses the queue `NODE_SYS_Q` to get the list of `SysNodes` from the base monitor running on the same host as the detector. The detector tries to create this queue until it is successful or for 10 times whichever is shorter. If after these attempts it is still unsuccessful it prints out the above message and exits with exit code 106.



*Action:*

Contact field support.

- (NOD, 13): ERROR: The RMS-CF-CIP mapping for SysNode <sysnode> to the CIP name has failed. Please verify all entries in /etc/hosts and /etc/cip.cf are correct and that CF and CIP are fully configured.

If there is no CIP entry corresponding to the SysNode <sysnode> in /etc/cip.cf, this message is the result and hvdet\_node exits with exit code 139.

*Action:*

Ensure that there is a corresponding CIP entry for the SysNode <sysnode> in /etc/cip.cf.

- (NOD, 16): ERROR: *detector*: failed to get information about RMS base monitor bm!

When the detector hvdet\_node finds that the RMS base monitor is not, it exits with exit code 142.

*Action:*

This might be due to the fact that hvdet\_node has been started independently of RMS.

- (NOD, 20): ERROR: *detector*: Cannot create socket: *errorreason*.

If there is a problem in the creation of an endpoint for communication between the detectors (*detector*) on the different hosts in the cluster, it manifests itself as a message in the switchlog and the detector exits with the exit code 111.

*Action:*

Contact field support.

- (NOD, 21): ERROR: *detector*: Failed to bind address to socket: *errorreason*.

If there is a problem in binding the endpoint of communication between the detectors (*detector*) on the different hosts in the cluster to a particular port, the result is this message with *errorreason* indicating the reason for this error. The detector then exits with exit code 130.

*Action:*

Contact field support.

- (NOD, 26): ERROR: *detector*: detector can't report resource state.

If the detector *detector* cannot report the state of the other SysNodes in the cluster to the base monitor running on the same host as the detector, this message is the result. This is most likely a problem with the queue when the detector is reporting the state.

*Action:*

Contact field support.

- (NOD, 28): ERROR: *detector*: SysNode list empty in hvdet\_node.

The hvdet\_node contacts the base monitor to get the list of SysNodes but if it just gets an empty list back in return, this message is the result. RMS then exits with exit code 129.

*Action:*

Contact field support.

- (NOD, 29): ERROR: 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 detector hvdet\_node can exit with any one of the exit codes 132, 133, 134, 135, 136, 137, 138 or 95.

*Action:*

Contact field support.

- (NOD, 30): ERROR: *detector*: message get doesn't work in hvdet\_node.

When the `hvdet_node` contacts the base monitor to get the list of `SysNodes`, if it finds that it could not get the list of `SysNodes` even after trying 10 times, this message is printed to the `switchlog`. This means that there is some problem with the message queues between the `hvdet_node` and the base monitor. RMS then exits with exit code 129.

*Action:*

Contact field support.

- (NOD, 31): ERROR: *detector*: nodename *nodename* not in NODELIST.

This message indicates a severe malfunction in RMS, when the detector *detector* cannot find the node *nodename* in its list of nodes.

*Action:*

Contact field support.

- (NOD, 34): ERROR: *detector*: Failed to call `osd select`: *error-reason*.

If the detector `hvdet_node` fails during the system call `select` while reading messages, this message is printed to the `switchlog` along with the reason *errorreason*. The detector then exits with exit code 131.

*Action:*

Contact field support.

- (NOD, 40): ERROR: *command*: `gethostbyname` returned NULL for host *hostname*.

If there is a problem in the detector when resolving a host *hostname*, this message is the result and the detector exits with exit code 114.

*Action:*

Ensure that you provide a valid host name.

- (NOD, 6): ERROR: Usage: *detector* -t *time\_interval*

If the detector `<detector>` has been provided a non-integer argument, this message is the result and the detector exits with exit code 103.

*Action:*

Provide an integer as the *time\_interval* for the detector *detector*.

- (NOD, 8): ERROR: Usage: *detector* -t *time\_interval* [-d] [-n]

If the argument -t *time\_interval* has not been provided for the detector <*detector*> or if an argument apart from -d or -n is used, this message is printed to the switchlog and the detector exits with exit code 103.

*Action:*

Follow the right usage for the invocation of the detector.

- (NOD, 9): ERROR: *detector*: Failed to open req\_queue

The detector *hvdet\_node* utilizes the queue *req\_q* for getting jobs from the base monitor. If there is some problem with the queue, this message is the result with the detector exiting with exit code 106.

*Action:*

Contact field support.

- (NOD, 10): ERROR: *detector*: Failed to open rep\_queue

The detector *hvdet\_node* utilizes the queue *rep\_q* to report the state of the other SysNodes in the cluster to the base monitor running on the same host as the detector. If there is a problem in sending the state over to the base monitor, this message is printed out and the detector exits with exit code 112.

*Action:*

Contact field support.

- (NOD, 11): ERROR: *service*: getservbyname returned NULL

If the detector has been unable to find the port at which the service <*service*> resides, this message is printed to the switchlog and the detector exits with exit code 126.

*Action:*

This is probably due to the absence of an entry for service <*service*> in */etc/services*.

- (NOD, 12): ERROR: *detector*: no NODE\_SYS\_Q

The detector `hvdet_node` uses the queue `NODE_SYS_Q` to get the list of `SysNodes` from the base monitor running on the same host as the detector. The detector tries to create this queue until it is successful or for 10 times whichever is shorter. If after these attempts it is still unsuccessful it prints out the above message and exits with exit code 106.

*Action:*

Contact field support.

- (NOD, 13): ERROR: The RMS-CF-CIP mapping for `SysNode` `<SysNode>` to the CIP name has failed. Please verify all entries in `/etc/hosts` and `/etc/cip.cf` are correct and that CF and CIP are fully configured.

If there is no CIP entry corresponding to the `SysNode` `<SysNode>` in `/etc/cip.cf`, this message is the result and `hvdet_node` exits with exit code 139.

*Action:*

Ensure that there is a corresponding CIP entry for the `SysNode` `<SysNode>` in `/etc/cip.cf`.

- (NOD, 16): ERROR: *detector*: failed to get information about RMS base monitor `bm`!

When the detector `hvdet_node` finds that the RMS base monitor is not, it exits with exit code 142.

*Action:*

This might be due to the fact that `hvdet_node` has been started independently of RMS.

- (NOD, 20): ERROR: *detector*: Cannot create socket: *error\_reason*

If there is a problem in the creation of an endpoint for communication between the detectors (*detector*) on the different hosts in the cluster, it manifests itself as a message in the switchlog and the detector exits with the exit code 111.

*Action:*

Contact field support.

- (NOD, 21): ERROR: *detector*: Failed to bind address to socket: *error\_reason*

If there is a problem in binding the endpoint of communication between the detectors (*detector*) on the different hosts in the cluster to a particular port, the result is this message with *error\_reason* indicating the reason for this error. The detector then exits with exit code 130.

*Action:*

Contact field support.

- (NOD, 26): ERROR: *detector*: detector can't report resource state

If the detector *detector* cannot report the state of the other SysNodes in the cluster to the base monitor running on the same host as the detector, this message is the result. This is most likely a problem with the queue when the detector is reporting the state.

*Action:*

Contact field support.

- (NOD, 28): ERROR: *detector*: SysNode list empty in *hvdet\_node*

The *hvdet\_node* contacts the base monitor to get the list of SysNodes but if it just gets an empty list back in return, this message is the result. RMS then exits with exit code 129.

*Action:*

Contact field support.

- (NOD, 29): ERROR: The RMS-CF interface is inconsistent and will require operator intervention. The routine *routine* failed with error code *error\_code* - *error\_reason*.

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

*Action:*

Contact field support.

- (NOD, 30): ERROR: *detector*: message get doesn't work in *hvdet\_node*

When the *hvdet\_node* contacts the base monitor to get the list of SysNodes, if it finds that it could not get the list of SysNodes even after trying 10 times, this message is printed to the switchlog. This means that there is some problem with the message queues between the *hvdet\_node* and the base monitor. RMS then exits with exit code 129.

*Action:*

Contact field support.

- (NOD, 31): ERROR: *detector*: nodename *nodename* not in NODELIST

This message indicates a severe malfunction in RMS, when the detector *detector* cannot find the node *nodename* in its list of nodes.

*Action:*

Contact field support.

- (NOD, 34): ERROR: *detector*: Failed to call `osd select`: *error\_reason*

If the detector *hvdet\_node* fails during the system call `select` while reading messages, this message is printed to the switchlog along with the reason *error\_reason*. The detector then exits with exit code 131.

*Action:*

Contact field support.

- (NOD, 40): ERROR: *hvdet\_node*: `gethostbyname` returned NULL for host *hostname*.

If there is a problem in the detector when resolving a host *hostname*, this message is the result and the detector exits with exit code 114.

*Action:*

Ensure that you provide a valid hostname.

- (QUE, 13): ERROR: RCP fail: *filename* is being copied.

If there is an attempt to copy the file with name *filename* when there is another copy in progress, this message results.

*Action:*

Ensure that concurrent copies of the same file do not occur.

- (SCR, 8): ERROR: Invalid script termination for controller *<controller>*.

The controller script is not correct or invalid.

*Action:*

Check the controller script.

- (SCR, 20): ERROR: The attempt to shutdown the cluster host *node* has failed: *error\_reason*.

The cluster node could not be killed because of any one of the following reasons:

1. Script exited with a non-zero status
2. Script exited due to signal caught
3. 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.

- (SWT, 4): ERROR: *object* is online locally, but is also online on *onlinenode*.

If the object *object* is online on more than one node, this message is the result.

*Action:*

Ensure that the object *object* is online on only one node in the cluster.

- (SWT, 20): ERROR: Could not remove host *<nodename>* from local priority list.



A node has left the cluster, but RMS was unable to remove the corresponding entry from its internal Priority List.

*Action:*

This is an internal problem in the program stack and memory management and should never occur. Contact field support.

- (SWT, 25): ERROR: *objectname*: outstanding switch request of dead host was denied; cluster may be in an inconsistent condition!

A node died during the processing of a switch request. The node that takes over the responsibility for that particular *userApplication* tried to proceed with the partly-done switch request, but another node does not agree.

*Action:*

This indicates a severe cluster inconsistency and should never occur. Contact field support.

- (SWT, 26): ERROR: *object*: dead host <*nodename*> was holding an unknown lock. Lock will be skipped!

This message appeared when the dead node <*nodename*> was holding a lock which is unknown to the new responsible node.

*Action:*

Allow time for the cluster to cleanup.

- (SWT, 45): ERROR: 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. Ensure the application finishes its processing before shutting down RMS.

- (SWT, 46): ERROR: hvshut aborted because modification is in progress.

The *hvshut* request was aborted because dynamic modification is in progress.

*Action:*

Do not shutdown RMS while dynamic modification is in progress. Wait until dynamic modification finishes before shutting down RMS.

- (SYS, 1): ERROR: Error on SysNode: *object*. It failed to send the kill success message to the cluster host: *node*.

When a cluster node is killed, the node that requested the kill must send a success message to the surviving nodes. This message appears in the switchlog when this message send fails.

*Action:*

Ensure the cluster and network conditions are such that the message can be sent across the network.

- (SYS, 8): RMS failed to shutdown the host *node* via a Shutdown Facility, no further kill functionality is available. The cluster is now hung.

This message appeared when the RMS was sending a kill request to the ShutDown Facility and did not get the elimination acknowledgement.

*Action:*

Refer to the *Cluster Foundation (CF) Configuration and Administration Guide* to determine what went wrong with the node elimination. Check the actual status of the remote node and invoke the appropriate `hvutil -u` or `hvutil -o` command to resolve the RMS hang state.

- (SYS, 13): ERROR: Since this host *<nodename>* has been online for no more than *time* seconds, and due to the previous error, it will shutdown now.

This message appears when the checksum of this node is different from the nodes in the cluster (one of the possible reasons).

*Action:*

Check the configuration in all the cluster nodes and verify that same configuration is running on all of them.

- (SYS, 14): ERROR: Neither automatic nor manual switchover will be possible on this host until *<detector>* detector will report offline or faulted.

This message appears when different configurations are encountered in a cluster where one node is Offline and the other is Online.

*Action:*

Check the configuration in all the cluster nodes and verify that same configuration is running on all of them.

- (SYS, 15): ERROR: 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:*

Ensure that the SysNode name is valid and restart RMS as needed.

- (SYS, 17): ERROR: 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 SysNode = `uname -n`RMS is intended to allow to use 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 *xxxRMS* and another one *xxx*, because *command xxx* could refer to either *xxxRMS* or *xxx*. Therefore any ambiguous SysNode name must not be allowed.

*Action:*

Use non-ambiguous SysNode name; remember, it is best to follow the naming conventions.

- (SYS, 48): ERROR: Remote host *<nodename>* replied the checksum *<remote\_checksum>* which is different from the local checksum *<local\_checksum>*. The sysnode of this host will not be brought online.

This message appears when the remote node *<nodename>* is running different configurations than the local node or different loads of RMS packages are installed on these nodes.

*Action:*

Ensure all the nodes are running the same configuration and the configuration is distributed on all nodes. Ensure that the same RMS package is installed on all nodes (same load).

- (SYS, 49): ERROR: Since this host *<nodename>* 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 node is different from the nodes in the cluster (one of the possible reasons).

*Action:*

Check the configuration in all the cluster nodes and verify that same configuration is running on all of them.

- (SYS, 50): ERROR: Since this host *<nodename>* has been online for no more than *time* seconds, and due to the previous error, it will shutdown now.

This message appears when the checksum of this node is different from the nodes in the cluster (one of the possible reasons).

*Action:*

Check the configuration in all the cluster nodes and verify that same configuration is running on all of them.

- (SYS, 84): ERROR: 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 nodes are still running.

*Action:*

Adjust the default timer by setting `RELIANT_SHUT_MIN_WAIT` to a value, which is large enough to allow a shutdown on all nodes. Check if the shutdown fails for internal problems (e.g. a failure of an `OfflineScript` causes a `userApplication` to fail to go Offline).

- (SYS, 90): ERROR: *hostname* internal WaitList addition failure! Cannot set timer for delayed detector report action!

**System Error.***Action:*

Contact System Administrator.

- (SYS, 93): ERROR: The cluster host *nodename* is not in the Wait state. The `hvutil` command request failed!

This message appears when the user issues the 'hvutil' command (`hvutil -o` or `-u`) and the cluster node *<nodename>* is not in Wait state.

*Action:*

Reissue `hvutil -{o, u}` only when the node is in a Wait state or don't issue it.

- (SYS, 94): ERROR: The last detector report for the cluster host *nodename* 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 node *<nodename>* 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 node is in a Wait state that has transitioned from Online state.

- (SYS, 97): ERROR: Cannot access the `NET_SEND_Q` queue.

When a new node comes `Online`, the other nodes in the cluster try to determine if the new node has been started with the `-C` option. The node which has just come `Online` uses the queue `NET_SEND_Q` to send the necessary information to the other nodes in the cluster. If this node is unable to access the queue `NET_SEND_Q`, this message is printed.

*Action:*

Contact field support.

- (SYS, 98): ERROR: Message send failed in `SendJoinOk`.

When a new node comes `Online`, the other nodes in the cluster try to determine if the new node has been started with the `-C` option. The node which has just come `Online` uses the queue `NET_SEND_Q` to send the necessary information to the other nodes in the cluster. If this node is unable to send the necessary information to the other nodes in the cluster, this message is printed.

*Action:*

Check if there is a problem with the network.

- (UAP, 1): ERROR: Request to go online will not be granted for application `<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 node `<SysNode>` is running a different configuration.

*Action:*

Ensure that the user is running the same configuration.

- (UAP, 5): ERROR: *object:* `cmp_Prio: list`.

This message is the result of the Priority list `list` having invalid entries.

*Action:*

Contact field support.

- (UAP, 6): ERROR: Could not add new entry to priority list.

This should never happen.

*Action:*

Contact field support.

- (UAP, 7): ERROR: Could not remove entries from priority list.

This should never happen.

*Action:*

Contact field support.

- (UAP, 8): ERROR: *object*: cpy\_Prio failed, source list corrupted.

This message appears when either the `PriorityList` is empty or the list is corrupted. This should never happen.

*Action:*

Contact field support.

- (UAP, 9): ERROR: *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 the result. The cluster may be in an inconsistent condition.

*Action:*

Contact field support.

- (UAP, 15): ERROR: *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. This should never happen.

*Action:*

Contact field support.

- (UAP, 16): ERROR: *object*::`SendUAppLockContract`: local host doesn't hold a lock -- Contract processing denied.

This message appears when the contract is processed by the node `<SysNode>` which does not have the lock for that application `<userApplication>` contract. This should never happen.

*Action:*

Contact field support.

- (UAP, 19): ERROR: *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): ERROR: *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): ERROR: *object* unlock processing failed, cluster may be in an inconsistent condition!

This message appears when the local node receives an UNLOCK contract but is unable to perform the follow up processing, which was committed in the contract.

*Action:*

Contact field support.

- (UAP, 23): ERROR: *object* failed to process UNLOCK contract.

A node was unable to propagate the received UNLOCK contract, e.g. because of networking problems or memory problems.



*Action:*

This message should never occur without an additional ERROR message specifying to the origin of the problem. Refer to the appropriate ERROR message.

- (UAP, 24): ERROR: Deleting of local contractUAP object failed, cannot find object.

This message appears when the local contract object has completed the contract and has sent it to the local node but the local object was not able to find it.

*Action:*

Contact field support.

- (UAP, 27): ERROR: *object* received a DEACT contract in state: *state*.

The correspondent userApplication on a remote node is in state DeAct, but the local userApplication is not. This should never happen.

*Action:*

Contact field support.

- (UAP, 28): ERROR: *object* failed to update the priority list. Cluster may be in an inconsistent state.

When the local node receives a contract for unlocking the nodes in the cluster with respect to a particular operation, if the local node finds that a particular node 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 the result.

*Action:*

This indicates an internal problem in memory management and should never occur. Contact field support

- (UAP, 29): ERROR: *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): ERROR: *object* received unknown contract.

This message appears when the application *<object>* is unable to process the contract because the contract is unknown. This should never happen.

*Action:*

Contact field support.

- (UAP, 33): ERROR: *object* unknown task in list of outstanding contracts.

This message appears when the application *<object>* finds a task in the list of outstanding contracts but is unable to process it. This should never happen.

*Action:*

Contact field support.

- (UAP, 35): ERROR: *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 is online and faulted.

*Action:*

Clear the inconsistency by the appropriate command (mostly `hvutil -c`).

- (UAP, 41): ERROR: 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): ERROR: found incorrect entry in status file:  
>*entry*<

This message appears when the `status_info` file has an incorrect entry in it. This should never happen except when 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.

- (US, 5): ERROR: The cluster host `nodename` is no longer reachable! Please check the status of the host and the Ethernet connection.

This message is a result of one cluster node detecting that the other node `nodename` which is part of the cluster is no longer reachable; in other words, this cluster node sees the other node `nodename` as faulted. This could be due to the fact that the other node `nodename` has gone down or there is some problem with the cluster interconnect.

*Action:*

Check if the node `nodename` is indeed dead. If not, check if there is a problem with the Ethernet connection.

- (US, 6): ERROR: RMS has died unexpectedly on the cluster host `nodename`!

When the detector on the local node detects that the node `nodename` has transitioned from `Online` to `Offline` unexpectedly, it prints this message to the switchlog and then it attempts to kill the node `nodename`.

*Action:*

Check the `syslog` on the node `nodename` to determine why the node has gone down.

- (US, 31): ERROR: FAULT REASON: Resource `resource_name` transitioned to a Faulted state due to a detector report.

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

*Action:*

No action required.

- (WLT, 1): ERROR: FAULT REASON: Resource *resource\_name*'s *script\_name* (*script*) has exceeded the ScriptTimeout of *timeout* seconds.

The detector script for the resource has exceeded the ScriptTimeout.

*Action:*

Ensure that timeout is large enough to execute the script.

- (WLT, 3): ERROR: Cluster host *nodename*'s script Shutdown Facility invoked via (*script*) has not finished yet. Operator intervention is required.

The Shutdown Facility that is killing node *nodename* has not terminated yet. Operator intervention may be required. This message will appear periodically (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 node being killed will not be considered killed.

*Action:*

Wait until the script terminates, or terminate the script using the `kill` command if the script cannot terminate on its own.

- (WLT, 5): ERROR: CONTROLLER FAULT: Controller *<object>* has propagated *<request>* request to its controlled application(s) *<applications>*, but the request has not been completed within the period of *<timeout>* seconds.

When the controller propagates its requests to the controlled applications, it is waiting for the completion of the request for a period of time specified in the controller's ScriptTimeout attribute. When the request is not completed within this period, controller faults.

*Action:*

Increase the controller's ScriptTimeout value, or fix the controller's scripts and/or scripts of the controlled applications, or repair resources of the controlled applications.

- (WRP, 1): ERROR: 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, 3): ERROR: Failed to execv: *command*.

This message could occur in any of the following scenarios:

1. If the detector cannot be started because RMS is unable to exec the detector process with the command *command*
2. When `hvcn -a` has been invoked, and the RMS base monitor cannot be started on the individual cluster nodes with the command *command*
3. When a script cannot be started because RMS is unable to exec the script process with the command *command*

*Action:*

Using the *errno* from the exit status of RMS, rectify the problem and restart RMS if necessary. Perform the following steps in order:

1. `echo $?`
2. `man -s 2 intro`

Compare the result from Step 1 with the error conditions specified in the manual page to find the name of the error.

3. Use the name of the error and then use the `man execv` command to look under the category `ERRORS` for the name of the error found in the previous steps.
- (WRP, 4): ERROR: Failed to create a process: *command*.

This message could occur in any of the following scenarios:

1. If the detector cannot be started because RMS is unable to create the detector process to execute the command *command*
2. When `hvcn -a` has been invoked and the RMS base monitor cannot be started on the individual cluster nodes with the command *command*
3. When 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 the *errno*. Refer to the manual page for `exec` to find out what the *errno* corresponds to in the context of `fork`.

*Action:*

Using the *errno* from the exit status of RMS, rectify the problem and restart RMS if necessary. Perform the following steps in order:

1. `echo $?`
2. `man -s 2 intro`

Compare the result from step 1 with the error conditions specified in the manual page to find the name of the error.

3. Use the name of the error and then do `man fork` and look for the error found in the previous steps to figure out the reason for RMS exiting.

- (WRP, 5): ERROR: No handler for this signal event `<signal>`.

There is no signal handler associated with this signal *signal*.

*Action*

Contact field support.

- (WRP, 9): ERROR: Socket open failed.

This message occurs if RMS is unable to create a datagram endpoint for communication.

*Action:*

Contact your System Administrator.

- (WRP, 11): ERROR: Message send failed, queue id `<queueid>`, process `<process>`, `<name>`, to host `<node>`.

RMS exchanges messages between nodes to maintain internode communication. This error occurs if the delivery of a message fails. This can occur if one or more nodes in the cluster are not active or if there is a problem with the network.

*Action:*

Perform the following steps:

1. Verify whether the other nodes in the cluster are alive or not; and if the other node or nodes in the cluster are not alive, check the switchlog for information regarding why RMS has died on the affected node or nodes. Perform these steps as follows:
    - a) Invoke `hvdisp -a`.
    - b) In the output of the preceding step, check if the `State` of any of the resources whose object type is `SysNode` are `Offline`. If so, that means that RMS is not running on that node.
    - c) Check the switchlogs of all the `SysNode` objects that are `Offline` to determine why RMS is not active on that node.
  2. If the other nodes that are part of the cluster are alive, then there is a problem with the network.
- (WRP, 12): ERROR: Failed to bind port to socket.

This could occur if RMS is unable to bind the endpoint for communication.

*Action:*

Contact your System Administrator.

- (WRP, 14): ERROR: 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 node in the cluster. When `hvdet_node` is started up, RMS sends it a list of the `SysNode` objects which 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 out.

*Action:*

Contact field support.

- (WRP, 15): ERROR: `gethostbyname(nodename)`: host name should be in `/etc/hosts`

When the `nodename` *nodename* specified as a `SysNode` does not have an entry in `/etc/hosts`, this message is printed out to the switchlog.

*Action:*

Correct the node name *nodename* to be an entry in `/etc/hosts`.

- (WRP, 16): ERROR: No available slot for host *nodename*

When RMS has run out of slots for the cluster interfaces (64), this message is printed along with the node name *nodename* for which this happened.

*Action:*

Contact field support.

- (WRP, 17): ERROR: Size of integer or IP address is not 4-bytes

This message should never appear.

*Action:*

Contact field support.

- (WRP, 23): ERROR: 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): ERROR: 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 the result.

*Action:*

Contact field support.

- (WRP, 25): ERROR: 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:*

Contact field support.



- (WRP, 28): ERROR: RMS monitor has encountered an irregular sequence of timer interrupts, off by *<offset>* seconds.

RMS 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 monitor which can cause a loss of High Availability.

*Action:*

Do not attempt to change system date/time by any significant value while RMS is running. Assure the system parameters allow for RMS to get enough CPU time to perform its operations during a high load.

- (WRP, 29): ERROR: RMS on the local host has received a message from host *node*, 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 node has received a message from node *node* whose address is not resolvable by the local node.

*Action:*

Make sure that the local node is able to resolve the remote node *node's* address by checking for any misconfigurations.

- (WRP, 30): ERROR: RMS on the local host has received a message from host *node*, but the local host is unable to resolve the sending host's address. This message will be dropped. Please check for any misconfiguration.

*Action:*

Make sure that the local node is able to resolve the remote node *node's* address by checking for any misconfigurations.

RMS on the local node has received a message from node *node* whose address is not resolvable by the local node.

- (WRP, 31): ERROR: RMS has received a message from host *node* with IP address *received\_ip*. The local host has calculated the IP address of that host to be *calc\_ip*. This may be due to a misconfiguration in `/etc/hosts`. Further such messages will appear in the switchlog.

The local node has received a message from node *node* with IP address *received\_ip*, which is different from the locally calculated IP address for that node.

*Action:*

Check `/etc/hosts` for any misconfigurations.

- (WRP, 32): ERROR: RMS has received a message from host *node* with IP address *received\_ip*. The local host has calculated the IP address of that host to be *calc\_ip*. This may be due to a misconfiguration in `/etc/hosts`.

The local node has received a message from node *node* with IP address *received\_ip*, which is different from the locally calculated IP address for that node. 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 misconfigurations.

- (WRP, 33): ERROR: Error while creating a message que 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): ERROR: Cluster host *node* is no longer in time sync with local node. Same operation of RMS can no longer be guaranteed. Further out-of-sync messages will appear in the syslog.

The time on *node* is not in sync with the time on the local node.

*Action:*

Sync the time on *node* with the time on the local node.

- (WRP, 35): ERROR: Cluster host *node* is no longer in time sync with local node. Same operation of RMS can no longer be guaranteed.

The time on the cluster node *node* differs significantly (> 25 seconds) from the local node.

*Action:*

Ensure that all the cluster nodes are in time sync.

## 5.12 Fatal error messages



The RMS product contains a feature for dynamically modifying the RMS configuration file. This feature is used both during RMS startup to establish the running configuration and by the currently unreleased RMS command `hvmmod` to perform runtime modification of the running configuration. The error messages for dynamic reconfiguration in this section apply to the base monitor internal use of the feature only.

- (ADC, 16): FATAL ERROR: 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_WARN_THRESHOLD`, and `HV_RCSTART` **have to be set in the `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): FATAL ERROR: 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 setup in the hvenv file, RMS prints this message and exits with exit code 1.

*Action:*

Ensure that all the local environment variables have been set to an appropriate value in the hvenv file.

- (ADC, 69): FATAL ERROR: RMS will not startup – 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.

- (ADM, 1): FATAL ERROR: 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): FATAL ERROR: RMS will not startup – 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:*

Ensure there are no errors in the configuration file based on the error messages printed prior to the above message in the switchlog.

- (BM, 3): FATAL ERROR: Usage: hvcm [-c config\_file] [-m] [-h time] [-l level] [-r count] [-w time] [-n]

If RMS has not been invoked in the right way because either some arguments were missing or haven't been used correctly, this message is printed out to the `switchlog` indicating the arguments. RMS exits with exit code 3.

*Action:*

Start RMS with the right arguments.

- (BM, 49): FATAL ERROR: 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): FATAL ERROR: The RMS-CF interface is inconsistent and will require operator intervention. The routine *routine* failed with errno *errno - error\_reason*

While setting up CF, if RMS encounters a problem in the routine *routine* which can either be `dlopen` or `dlsym`, it exits with exit code 95 or 94 respectively. The *error\_reason* gives the reason for the error.

*Action:*

Contact field support.

- (BM, 58): FATAL ERROR: Not enough memory -- RMS cannot continue its operations and is shutting down

This is a generic message that is printed out to the `switchlog` before RMS discontinues its functioning because it does not have enough memory to operate.

*Action:*

Contact field support.

- (BM, 67): FATAL ERROR: 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 the `switchlog` explains why RMS has not been able to write down the configuration file. Please correct the node environment according to the description, or contact field support.

- (BM, 82): FATAL ERROR: A message to host `<remotenode>` failed to reach that host after `<count>` delivery attempts. Communication with that host has been broken. Therefore, RMS monitor on this host `<localnode>` 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 communication between two nodes is reliably established. Then restart local RMS monitor.

- (BM, 89): FATAL: The SysNode length is *length*. This is greater than the maximum allowable length of *maxlength*. RMS will now shutdown.

The SysNode length is greater than the maximum allowable length.

*Action:*

Ensure that the length of the SysNode is smaller than *maxlength*.

- (CML, 14): FATAL ERROR: ###ERROR: Unable to find or Invalid configuration file.###

```
#####CONFIGURATION MONITOR exits !!!!!#####
```

If the configuration file specified for RMS is non-existent, then this message is printed to the `switchlog` and RMS exits with exit code 1.

*Action:*

Specify a valid configuration file for RMS to function.

- (CMM, 1): FATAL ERROR: Error establishing outbound network communication

If there is an error in creating outbound network communication, this message is the result and RMS exits with exit code 12.

*Action:*

System error. Contact field support.

- (CMM, 2): FATAL ERROR: Error establishing inbound network communication

If there is an error in creating inbound network communication, this message is the result and RMS exits with exit code 12.

*Action:*

System error. Contact field support.

- (CMM, 3): FATAL ERROR: create queue error NODE\_SYS\_Q

The `NODE_SYS_Q` is used by the RMS base monitor to communicate the list of `SysNode` objects to `hvdet_node`. If there is a problem creating this queue for some reason, RMS exits with exit code 12.

*Action:*

Contact field support.

- (DET, 8): FATAL ERROR: 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 the result and RMS exits with exit code 12.

*Action:*

Contact field support.

- (DET, 9): FATAL ERROR: 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 the result and RMS exits with exit code 12.

*Action:*

Contact field support.

- (DET, 16): FATAL ERROR: Cannot create gdet queue of kind *gkind*

Each of the generic detectors has a message queue which 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 the result and RMS exits with exit code 12.

*Action:*

Contact field support.

- (DET, 18): FATAL ERROR: 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.

- (INI, 4): FATAL ERROR: InitScript does not have execute permission.

InitScript exists, but cannot be executed.

*Action:*

Make InitScript executable.

- (INI, 7): FATAL ERROR: *SysNode* must be in your configuration file

If the local *SysNode SysNode* is not part of the configuration file, this message is the result and RMS exits with exit code 23.

*Action:*

Ensure that the local *SysNode SysNode* is part of the configuration file.

- (INI, 10): FATAL ERROR: InitScript has not completed within the allocated time period of *timeout* seconds.



InitScript was still running when the time period allocated for its execution expired. The timeout period is the least of the values defined in the environmental variable `SCRIPTS_TIME_OUT` from `hvenv` file, or 300.

*Action:*

Increase the timeout value, or correct the conditions that lead to timeout during script execution.

- (INI, 11): FATAL ERROR: 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 node condition for InitScript to be able to startup.

- (INI, 12): FATAL ERROR: InitScript returned non-zero exit code *exitcode*.

InitScript completed with a non-zero exit code *<exitcode>*.

*Action:*

Correct the erroneous node condition for InitScript to be able to return a zero exit code, or fix the InitScript itself.

- (INI, 13): FATAL ERROR: InitScript has been stopped.

InitScript has been stopped.

*Action:*

Correct the erroneous node condition for InitScript to run without stopping, or fix the InitScript itself.

- (INI, 14): FATAL ERROR: InitScript has been abnormally terminated.

InitScript has been abnormally terminated.

*Action:*

Correct the erroneous node condition for InitScript to run without stopping, or fix the InitScript itself.

- (MIS, 4): FATAL ERROR: The locks directory *directory* cannot be cleaned of all old locks files

The various RMS commands like `hvdisp`, `hvswitch`, `hvutil`, `hveject`, `hvjoin`, 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 the result. RMS exits with exit code 99.

*Action:*

Ensure that the locks directory *directory* exists.

- (QUE, 1): FATAL ERROR: 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 the result and RMS exits with exit code 3.

*Action:*

Contact field support.

- (QUE, 2): FATAL ERROR: Read message failed in ADMIN\_Q

This message is the result of the RMS base monitor being unable to extract a message of the ADMIN\_Q which is used for communication between the utilities and RMS. RMS then exits with exit code 3.

*Action:*

Contact field support.

- (QUE, 5): FATAL ERROR: Network message read failed

If there is a problem reading a message over the network, this error is the result and RMS exits with exit code 3.

*Action:*

System error. Contact field support.

- (QUE, 6): FATAL ERROR: Network problem occurred

This message is the result of a network problem occurring when transferring messages.

*Action:*

System error. Contact field support.

- (QUE, 11): FATAL ERROR: 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): FATAL ERROR: Error status in DET\_REP\_Q: *status*

This message is the result of the RMS base monitor having a problem with the queue DET\_REP\_Q which is used by the different detectors to report their state. RMS then exits with exit code 15.

*Action:*

Contact field support.

- (SCR, 4): FATAL ERROR: Failed to create a detector request queue for detector *detector\_name*

If a detector request queue could not be created for detector *detector\_name*, this message is the result and RMS exits with exit code 12.

*Action:*

System problem. Contact field support.

- (SCR, 5): FATAL ERROR: REQUIRED PROCESS RESTART FAILED: Unable to restart *detector*. Shutting down RMS

If the detector *detector* could not be restarted, this message is the result with RMS shutting down with exit code 14. The restart could have failed for any of the following reasons:

- If the detector needs to be restarted more than three times in 1 minute

- If there is a problem with memory allocation within RMS

*Action:*

Contact field support.

- (SCR, 10): FATAL ERROR: 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` in `henvv`. For some reason, if this `InitScript` fails (like exiting with a non-zero code, getting a signal, etc.), then this message is printed and RMS shuts down with exit code 56.

*Action:*

Contact field support.

- (SCR, 12): FATAL ERROR: 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 a valid `Online`, `Offline`, `Faulted`, `Standby`, or `NoReport`, this message is the result with RMS exiting with exit code 8.

*Action:*

Ensure that the detector only reports valid states such as `Online`, `Offline`, `Faulted`, `Standby`, or `NoReport`.

- (SCR, 13): FATAL ERROR: `ExecScript`: Failed to exec script `<script>` for object `<objectname>`: errno `errno`

If RMS has been unable to execute a script `<script>` for some particular object `<objectname>`, RMS prints this error message. The errno `errno` provides a much better idea as to why this execution has failed. RMS exits with exit code 8.

*Action:*

Perform the following:

1. `man -s 2 intro`
2. Check the errno `errno` with the manual page to determine the reason.

- (SCR, 15): FATAL ERROR: node\_sys\_q cannot be accessed

The queue `node_sys_q` is used by the detector `hvdet_node` to get the list of the `SysNodes` from the RMS base monitor, if there is some problem with this queue, this message is printed and RMS exits with exit code 12.

*Action:*

Contact field support.

- (SCR, 18): FATAL ERROR: Message send failed to node\_sys\_q.

The RMS base monitor uses the queue `node_sys_q` to send the list of `SysNodes` to `hvdet_node` after `hvmmod` (the initial one on startup or the subsequent ones when `hvmmod` has been invoked explicitly). If RMS is unable to send this information to `hvdet_node`, this message is printed and RMS exits with exit code 2.

*Action:*

Contact field support.

- (SYS, 33): FATAL ERROR: The RMS cluster host `<nodeName>` 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` which searches the file `/etc/hosts` to get information about the node `nodeName` is unable to find a valid entry for it, this message is printed and RMS exits with exit code 114.

*Action:*

Ensure that the nodename `nodeName` has a valid entry in `/etc/hosts` and restart RMS.

- (SYS, 52): FATAL ERROR: 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 the result and RMS exits with exit code 12.

*Action:*

Contact field support.

- (UAP, 36): FATAL ERROR: *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 *object*, if a double fault occurs, then RMS will exit with exit code 96 on that node.

*Action:*

Contact field support.

- (US, 1): FATAL ERROR: RMS will not startup – fatal errors in configuration file.

During RMS startup, errors were found in the configuration file causing RMS to not startup.

*Action:*

Correct the errors in the configuration file.

- (US, 42): FATAL ERROR: A State transition error occurred. See the next message for details.

A state transition error occurred in the course of RMS state transitions. Details of the error are printed in the subsequent lines.

*Action*

Save the error description and contact field support.

## 5.13 Additional error messages



The RMS product contains a feature for dynamically modifying the RMS configuration file. This feature is used both during RMS startup to establish the running configuration and by the currently unreleased RMS command `hvmod` to perform runtime modification of the running configuration. The error messages for dynamic reconfiguration in this section apply to the base monitor internal use of the feature only.

The following messages are printed on `stderr`.

- Usage: hvassert [-h SysNode] [-q] -s resource\_name resource\_state | [-h SysNode] [-q] -w resource\_name resource\_state seconds | (-f|-m) list | (resource\_name resource\_state [[or|OR] resource\_state]...)... [seconds]

If the utility `hvassert` has been invoked in a way that does not conform to its expected usage, this message is printed and the utility exits with exit code 6.

*Action:*

Follow the usage specified above.

- Usage: hvdisp {-a | -c | -h | -i | -l | -m | -n | -S resource\_name [-u | -c] | -T resource\_type [-u | -c] | -u | resource\_name } [-o out\_file]

An attempt to use the `hvdisp` utility in a way that does not conform to the expected usage leads to this message and the utility exits with exit code 6.

*Action:*

Follow the expected usage for the utility.

- FATAL ERROR: RMS has failed to start. Error message:  
*error\_reason*

Internal error

*Action:*

Contact field support.

- File open failed (*filename*): *error\_reason*.

If the file *filename* that is used by the `hvassert` utility to communicate with the RMS base monitor could not be opened, this message is the result, along with the reason *error\_reason* for this failure. `hvassert` then exits with exit code 5.

*Action:*

Contact field support.

- Resource does not exist - *resource\_name*.

If there is an attempt to perform a `hvassert` on a resource which is not part of the RMS resource graph, this message is printed and `hvassert` then exits with exit code 10.

*Action:*

Ensure that a resource exists before performing an `hvassert` on it.

- Remote system is not online.

Trying to perform a `hvassert` on an object on a remote node which does not have RMS running causes this message and `hvassert` exits with exit code 10.

*Action:*

Ensure that the remote system has RMS running before performing an `hvassert`.

- Remote host does not exist – *host*.

If the SysNode *sysNode* specified as `hvassert -h host ...` is not part of the RMS resource graph, this message is printed and `hvassert` exits with exit code 10.

*Action:*

Ensure that the remote nodename specified for `hvassert` exists.

- `hvassert: bad state: state`.

If `hvassert` is performed for a state *state* which is not among the states that can be asserted, this message is the result and `hvassert` exits with exit code 1.

*Action:*

Ensure that the state specified for `hvassert` is assertable.

- `hvassert: bad timeout: timeout`.

If the timeout specified for the `hvassert` command is not a number, this message is the result and the utility exits with exit code 1.

*Action:*

Specify a number for the timeout value of `hvassert`.



- Remote host `<nodename>` is not Online.

When performing `hvassert`, if the remote node `nodename` is not Online, this message is printed and `hvassert` exits with exit code 1.

*Action:*

Ensure that the remote node is Online before performing `hvassert`.

- `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:*

No action required.

- Dynamic modification is in progress, can't assert states.

It is not possible to perform an `hvassert` when dynamic modification is in progress.

*Action:*

Perform `hvassert` after dynamic modification finishes.

- Assertion condition failed.

If the `hvassert` fails while using `-f` or `-F` options, this message is printed and `hvassert` exits with exit code 1.

*Action:*

No action required.

- The length of return message from BM is illegal (`actual_length` actual `expected_length` expected).

When the `hvassert` utility expecting a return message from the base monitor receives a message of length `actual_length` when it is expecting a message of length `expected_length`, this message is printed and `hvassert` exits with exit code 5.

*Action:*

Contact field support.

- Too many asserted objects, *maximum* is the max.

Any attempt to assert on a number of objects which is greater than the *maximum* will cause this message to be printed.

*Action:*

Ensure that the number of asserted objects is less than the maximum.

- Usage: `hvconfig -l | -o config_file`

An attempt to use the `hvconfig` utility in a way that does not conform to the expected usage leads to this message and the utility exits with exit code 6.

*Action:*

Follow the expected usage for the utility.

- Usage: `hvdump {-f out_file} | -t wait_time}`

An attempt to use the `hvdump` utility in a way that does not conform to the expected usage leads to this message and the utility exits with exit code 6.

*Action:*

Follow the expected usage for the utility.

- It may take few seconds to do Debug Information collection.

The `hvdump` utility dumps out the information regarding the resource graph. It prints out the above message when it is collecting the information.

*Action:*

No action 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 proprietary files you do not want included, please exit now. Do you want to proceed? (yes = continue / no = quit)

This message is printed out on executing `hvdump -E` and will collect the necessary information only if the answer to the above question is *yes*.

*Action:*

No action required.

- **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:*

No action required.

- Root access required to start `hvcn`

To start RMS, the user must have root access.

*Action:*

Login as root and try `hvcn`.

- 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 `hvcn` manual page for the correct syntax and usage.

- RMS has failed to start!

The options `-a` and `-s` are incompatible and may not be specified both

This message appears when you use the options `-a` and `-s` simultaneously to start RMS.

*Action:*

Check the man page for `hvcn` to get the format.

- RMS has failed to start!

hvcn has been invoked without specifying a configuration with the `-c` attribute, but with specifying other commandline options. This may cause ambiguity and is therefore not possible. Please specify the entire commandline or use "hvcn" without further options to run the default configuration

This message appears when you start RMS without the `-c` option and specifying other command line options.

*Action:*

The user can start RMS with entire command line including `-c` option and the `-c` option should be the last command line option followed by the configuration file name. The user can also start RMS without specifying the any command line options which will start the default configuration on the system.

- RMS has failed to start!

multiple entries in the RMS default configuration file "*config\_file*"

The user is not allowed to start RMS if there are multiple entries in the default configuration file `config.us`.

*Action:*

Remove all the obscure entries in the RMS default configuration file and have only one valid configuration in it.

- RMS has failed to start!

invalid entry in the RMS default configuration file *config\_file*

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 as follows:

1. *configname*
2. `hvcn <options> -c configname`

Refer to `hvcn` manual page for details on valid options in format b.

*Action:*

Remove all invalid entries in the RMS default configuration file. Refer to the `hvc` manual page.

- RMS has failed to start!

the number of arguments specified at the RMS default configuration file "`config_file`" overrides the internal buffer of the RMS start utility

This message appears when you try to start RMS using the RMS default configuration file but are 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 configuration file. Check the manual page for `hvc` to get the required options to start RMS.

- RMS has failed to start!

didn't find a valid entry in the RMS default configuration file "`config_file`"

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 which the user wants to start.

- 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 confirmation to activate the new configuration file. If the response is no, then this message is the result.

*Action:*

No action required.

- RMS has failed to start!

RELIANT\_HOSTNAME is not defined in the RMS environment

This should never happen.

*Action:*

Ensure that the RMS environmental variable RELIANT\_HOSTNAME wasn't set erroneously to "" or explicitly unset in `hvenv.local`.

- Error while starting up `bm` on the remote host `<targetnode>`:  
*error\_reason*

When `hvcn` is invoked with the `-s` option to start RMS on a remote node `<targetnode>`, if there is a problem in starting up RMS on the remote node, this message is the result along with the reason for the problem `<error_reason>`.

*Action:*

Take action based on the reason for the problem and reissue `hvcn -s`.

- Error becoming a real time process: *error\_reason*

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 *error\_reason*, then this message is the result.

*Action:*

Take action based on the reason.

- Error setting up real time parameters: *error\_reason*

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 the result along with the reason *error\_reason* for the problem.

*Action:*

Take action based on the reason.

- Error while starting up local `bm`: *error\_reason*

Error while starting up local `bm`: *error\_reason*

*Action:*

Take action based on the reason.

- Usage: hvcm [-V] [-a] [-s targethost] [-c config\_file] [-m] [-h time] [-l level] [-r count] [-R] [-w time]

Usage is not correct.

*Action:*

Check hvcm manual page for correct usage.

- MESSAGE The configuration file *non\_default\_config\_file* has been specified as option argument of the `-c` option, but the Wizard Tools activated configuration is *default\_config\_file* (see *default\_config\_file*). The base monitor will not be started. The desired configuration file should be re-activated using the Wizard Tools `hvw` command.

This message is shown when you try to start RMS with a configuration different from the configuration present in the RMS default configuration file. The base monitor is not started. You will need to either change the default configuration file by re-activating the configuration via the RMS Wizards Tools `hvw` command or specify the proper option argument for the `-c` option.

*Action:*

Correct the default configuration by activating the specified configuration file using the RMS Wizards Tools or specify the proper option argument to the `-c` option.

- Starting RMS on remote host *node* now

This message will be printed when RMS is being started on the remote node *node*.

*Action:*

No action required.

- 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 nodes in the cluster.

*Action:*

No action required.

- Failed to open pipe.

If RMS is unable to open a pipe for communication, this message is the result and RMS exits with exit code 1.

*Action:*

Contact field support.

- Fork failed.

If RMS is unable to fork a process, it prints this message and exits with exit code 1.

*Action:*

Contact field support.

- 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>*.

If RMS is unable to exec the hvenv file *hvenvfile*, this message is printed out.

*Action:*

Contact field support.

- Starting Reliant Monitor Services now

When RMS is starting up, this message is printed out.

*Action:*

No action required.

- Usage: hveject -s host



An attempt to use the `hveject` utility in a way that does not conform to the expected usage leads to this message and the utility exits with exit code of 2 or 6, depending on one of the following conditions:

1. If an unknown option is used, the exit code is 2.
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`

An attempt to use the `hvjoin` utility in a way that does not conform to the expected usage leads to this message and the utility exits with exit code of 2 or 6, depending on one of the following conditions:

1. If an unknown option is used, the exit code is 2.
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]`

An attempt to use the `hvlogclean` utility in a way that does not conform to the expected usage leads to this message and the utility exits with exit code 6.

*Action:*

Follow the expected usage for the utility.

- `RELIANT_PATH` is not defined

When the `hvlogclean` utility is invoked without the `-d` option, it needs the value of the environment variable `RELIANT_PATH`, to get to the `hvloginit` script. If the value of the variable cannot be found, this message is the result and the utility exits with exit code 6.

*Action:*

Ensure that the environment variable `RELIANT_PATH` was not unset and is set to the appropriate value.

- `<command>` failed with exit code `exit_code`

When the `hvtlogclean` 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 `exit_code` and then the utility exits with exit code 6.

*Action:*

Take action based on the exit code `exit_code`.

- `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:*

Ensure 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 the result and the utility exits with exit code 2.

*Action:*

Ensure that RMS is running before invoking the different utilities.

- `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 the result and the utility exits with exit code 3.

*Action:*

Contact field support.

- `directory`: cannot put message in queue

The various RMS commands like `hvdisp`, `hvswitch`, `hvutil`, `hveject`, `hvjoin`, 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 the result. RMS exits with exit code 99.

*Action:*

Ensure that the locks directory *directory* exists.

- *command*: could not create a pipe

If the utility *command* could not open the tty to be written to, this message is the result and the utility exits with exit code 7.

*Action:*

Contact field support.

- *command*: file already exists

When `hvdisp -o` has been invoked by the user, if the output file that has been specified as an argument already exists, this message is the result and `hvdisp` exits with exit code 6.

*Action:*

Specify a filename that does not already exist as the argument to `hvdisp -o`.

- The system call *systemcall* could not be executed: *errmsg*

While performing a `hvdump`, if the *system call* could not be executed because of *errmsg*, this message is the result and `hvdump` exits with exit code 7.

*Action:*

Take action based on the *errmsg*.

- The file *filename* could not be opened: *errmsg*

While performing a `hvdump`, if the file *filename* could not be opened because of *errmsg*, this message is the result and `hvdump` exits with exit code 8.

*Action:*

Take action based on the *errmsg*.

- hvmod cannot get list of resources via *<command>* from hvcm.

The RMS Wizards rely on hvmod for dynamic modification. If there is a problem executing command *command*, this message is the result and hvmod exits with exit code 15.

*Action:*

Contact field support.

- hvmod 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 the result and hvmod is aborted. hvmod then exits with exit code 1.

*Action:*

Contact field support.

- Modification file name is missing on the command line, usage: hvmod [-i] [-l] -f config\_file.us | -E | -L | [-i] [-l] -c "modification directives"

When the hvmod utility is invoked with an option that does not conform to its expected usage this message is the result and the utility exits with exit code 2.

*Action:*

Follow the expected usage for the utility.

- Usage: hvmod [-i] [-l] -f config\_file.us | -E | -L | [-i] [-l] -c "modification directives"

If the hvmod utility is invoked in any one of the following ways, this message is the result and the hvmod exits with exit code 6:

1. If hvmod is invoked without any options
2. If hvmod is invoked with the -l or -i options but with arguments when none are expected

*Action:*

Follow the expected usage for the utility.

- Name of the modification file is too long.

If the name of the modification file specified as an argument through the `-f` option or the modification directives specified via the `-c` option are greater than 113, this message is printed and `hvmmod` exits with exit code 4.

*Action:*

Ensure that the arguments specified via `-f` and `-c` options are not too long.

- Can't open modification file.

When `hvmmod` is invoked with the `-c` option, it utilizes a temporary file, if this file cannot be opened for writing, this message is the result and `hvmmod` exits with exit code 1.

*Action:*

Contact field support.

- Too many arguments, usage: `hvmmod -E`

The `hvmmod` utility does not expect any arguments when invoked with the `-E` option. If not, `hvmmod` exits with exit code 1.

*Action:*

Ensure that `hvmmod -E` is not invoked with any arguments.

- Usage: `hvrncp localfile node:remotefile`

This message is the result of either of the following conditions:

1. If the number of arguments specified is not equal to 2
2. If the second argument is not specified in the form *node:remotefile*

Then `hvrncp` exits with exit code 6

*Action:*

Follow the intended usage of `hvrncp` as specified above.

- `localfile filename` does not exist or is not an ordinary file

If the local file specified as an argument to `hvrpcp` does not exist or if it is not a regular file, `hvrpcp` exits with exit code 7.

*Action:*

Ensure that the local file exists and is an ordinary file.

- Could not open `localfile` or could not create temporary file `filename`

If during `hvrpcp`, the local file cannot be opened for reading or the temporary file `filename` cannot be opened for writing, this message is printed and `hvrpcp` exits with exit code 7.

*Action:*

Check the permissions on the local file to make sure that it is readable.

- `rms is dead`

The `hvrpcp` 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 node.

- `timed out! Most likely rms on the remote host is dead.`

While performing `hvrpcp`, if the command times out because the base monitor on the local node has not received an acknowledgement from the base monitor on the remote node, the most probable reason is that the RMS on the remote node is dead.

*Action:*

Ensure that the RMS on the remote node is running.

- Usage: `hvshut {-f | -L | -a | -1 | -s SysNode}`

If the usage of `hvshut` does not conform to the expected usage as specified in the above message, the `hvshut` utility exits with the exit code 6.

*Action:*

Follow the intended usage of `hvshut` as specified above.

- 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 determine the value of this variable, this message is printed.

*Action:*

Ensure that RELIANT\_PATH is set to an appropriate value.

- *time\_stamp*:NOTICE: User has been warned of 'hvshut -f' and has elected to proceed.

When the user invokes `hvshut -f` and elects to proceed with the command, then this message is printed to confirm that `hvshut -f` is being invoked.

*Action:*

No action required.

- NOTICE: User has been warned of 'hvshut -L' and has elected to proceed.

When the user invokes `hvshut -L` and elects to proceed with the command, then this message is printed to confirm that `hvshut -L` is being invoked.

*Action:*

No action required.

- Forced shut down on the local cluster host!

When the detector restarts the base monitor, it prints this message before proceeding.

*Action:*

No action required.

- 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-establishment includes restart of RMS on the shut down host.

Do you wish to proceed? (yes = shut down RMS / no = leave RMS running).

This message asks you if you want to proceed with `hvshut -f`. **yes** means proceed with `hvshut -f`. **no** means discontinue with `hvshut -f`.

*Action:*

No action required.

- Shutdown of RMS has been aborted.

When the user invokes `hvshut -L`, the `hvshut` utility asks for a confirmation, if the answer is **no** then `hvshut -L` is aborted, and this message is printed out.

*Action:*

No action required.

- WARNING

The '-L' option of the `hvshut` command will shut down the RMS software without bringing down any of the applications.

In this situation, it would be possible to bring up the same application on another node in the cluster which \*may\* cause data corruption.

Do you wish to proceed? (yes = shut down RMS / no = leave RMS running).

This message asks you if you want to proceed with `hvshut -L`. **yes** means proceed with `hvshut -L`. **no** means discontinue with `hvshut -L`.

*Action:*

No action required.

- Sending *data* to *resource*.

This message is printed when logging is turned on and *data* is being sent to the object *resource*.

*Action:*

No action required.

- Usage: `hvswitch [-f] userApplication [SysNode]`



If an unknown option is used with the `hvs witch` utility or if there are more than 2 arguments specified for `hvs witch`, it exits with exit code 6.

*Action:*

Follow the intended usage of the utility.

- Usage: `hvutil {-a | -d | -D | -f | -c | -s} userApplication | -t N resource | -L{0|1} resource | {-o | -u} SysNode | -l level | -z [all | resource...] | {-w | -W} [ SysNode ] | -i {all | userApplication} | -r`

This message could appear in any one of the following situations:

1. `hvutil -u` is invoked with more than 1 argument. Exit code 7.
2. `hvutil` is invoked without any options or arguments. Exit code 7.
3. `hvutil` is invoked with an illegal option. Exit code 7.
4. `hvutil -i` is used without an argument. Exit code 13.
5. `hvutil -r` is used with an argument. Exit code 14.
6. `hvutil {-w | -W}` is invoked with more than one argument. Exit code 9.
7. `hvutil -n` is invoked with `NoConfirm` as the only argument. Exit code 5.

*Action:*

Follow the intended usage of `hvutil`.

- `hvutil`: Detector time period must be greater than *minimum\_time*.

If the detector time period specified as an argument with `hvutil -t` is less than *minimum\_time*, `hvutil` is aborted and exits with exit code 5.

*Action:*

Invoke `hvutil` with a time period that is greater than *minimum\_time*.

- `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:*

Ensure that the detector time period is an integer.

- WARNING

Data corruption may occur, if the SysNode referred to as option-argument of the `-u` option hasn't been completely deactivated.

Do you wish to proceed? (default: no) [yes, no]:

**This is printed if `hvutil -u` has been invoked. `yes` means proceed with `hvutil -u`. `no` means discontinue with `hvutil -u`.**

*Action:*

No action required.

- WARNING

The `-k` option is intended to be used as an test tool for engineering ONLY

Do you wish to proceed? (default: no) [yes, no]:

**This message is the result of invoking `hvutil -k` with the argument `engkilltest`. Answering with a `yes` causes the command to proceed and to abort if `no` is typed in.**

*Action:*

No action required.

- `hvutil`: debug option must be a number -- 1 for on, 0 for off.

**When `hvutil -L` has been invoked with a log level that is not 0 or 1, this message is the result and it exits with exit code 6.**

*Action:*

Specify a valid logging level of 0 or 1 for the utility.

- `hvutil`: RMS is not running on `<targetnode>`

**Printed when `hvutil -A targetnode` is called, indicating that RMS is not running on the named node.**

*Action:*

No action required.

- `hvutil: RMS is running on <targetnode>`

Printed when `hvutil -A targetnode` is called indicating that RMS is running on the named node.

*Action:*

No action required.

- `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 nodes for `rmshb` using `tcp`.

- `hvutil: Failed to allocate socket`

Failed to allocate a socket to communicate with a remote node.

*Action:*

Contact field support to determine the cause.

- `hvutil: Could not determine IP address of <targetnode>`

The name of the cluster node could not be resolved to an IP address.

*Action:*

Add an entry for `targethost` into the `/etc/hosts` file of all cluster nodes.

- `hvutil: Could not determine if RMS is running on <targetnode>, errno exitcode`

Printed when `hvutil -A targetnode` is called indicating that the command failed to ascertain whether or not RMS is running on `targetnode`. The *exitcode* indicates a value in `/usr/include/sys/errno.h`.

*Action:*

Depends on the *exitcode* value.

## 5.14 RMS troubleshooting

RMS is a stable product. In case of problems it prints out meaningful error messages, which will help you in troubleshooting. Additionally, the following is provided to help solve problems with running RMS:

- 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`, so 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.

*Action:*

Verify the problem by calling `hvdisp -a` on the remote nodes to find out the actual configuration files. Compare the checksum of these configuration files.

If the base monitor does not shut down on its own, but keeps running because one of the above conditions is not true, the system administrator may need to do the following:

1. Shut down certain base monitors.
2. Find out which configuration to run.

3. Distribute this file with `hvdist`.
  4. Stop and restart RMS on the entire cluster so that all cluster nodes run 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 the other (at least some other) cluster nodes.

*Action:*

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

Call `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 (same format), so RMS will invoke a `gethostbyaddr()`. This normally does not do any harm, but in some strange cases this seems to cause Solaris to 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.



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 calling `hvdisp -T SysNode` to see the states of all `SysNode` objects.

If you verify that a `SysNode` is in a pending Wait state, call `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 detector cycle time can be changed from its default value by using the `-w` option in `hvcn` command as `hvcn -w <n> -c <config_file>` where *n* is the new detector cycle time. This value must be greater than the value of `HV_CONNECT_TIMEOUT`.

---

## 6 Appendix—Object types

Table 7 contains a list of all object types that are supplied with RMS. The middle column lists the attributes that must be specified or are recommended for the object type in the object configuration file definition.

Type	Required Attributes	Description
andOp	HostName for direct children of a userApplication object	Object that is associated with its children by a logical AND operator. Define this type of object if all children have to be online or offline at the same time.
controller	Resource	Object within a userApplication that controls one or more userApplication objects
gResource	rKind, rName	Custom (generic) object
ENV	No attributes required	Object containing clusterwide (global) environment variables
ENVL	No attributes required	Object containing machine-specific (local) environment variables

Table 7: Object types

Type	Required Attributes	Description
orOp	No attributes required	Object associated with its children by a logical OR operator. Define this type of object if at least one child has to be online at all times.
SysNode	No attributes required	Node object; required. Only type <code>userApplication</code> can be defined for the children.
userApplication	No attributes required	User application; required for every application. Only <code>SysNode</code> parents are allowed. The attribute <code>HostName</code> must be set for children.


Table 7: Object types



---

## 7 Appendix—Attributes

Some object types require specific attributes for RMS to monitor that object type. Table 8 lists the attributes that are required for each object type.

 Not all of these attributes are modifiable by the user. The values of some attributes are set by the RMS Wizards internally.

Attribute	Possible Values	Description
Affiliation	Any string	Valid for resource objects. No functional meaning within RMS. Used within the RMS Wizards to indicate membership of sub-applications. No default value.
AutoRecover	0, 1	Valid for resource objects. Executes the online script for an object if the object becomes faulted while in an <code>Online</code> state. If the object is able to return to the <code>Online</code> state, the fault is recovered. <b>Default:</b> 0.
AutoRecoverCleanup	0, 1	Valid for controller objects. If set to 1, and <code>AutoRecover</code> is 1, then a faulted controlled <code>userApplication</code> object is requested to go <code>Offline</code> before recovering. If 0 and <code>AutoRecover</code> is 1, then a faulted controlled <code>userApplication</code> object recovers without going <code>Offline</code> . <b>Default:</b> 1.

Table 8: RMS attributes

Attribute	Possible Values	Description
AutoStartUp	0, 1	Automatically switches a userApplication object Online on the SysNode with the highest priority when RMS is started. Set to either 0 for no or 1 for yes. <b>Default:</b> 0.
AutoSwitchOver	No HostFailure Resource- Failure Shutdown	Configures a userApplication object for automatic switchover if it becomes faulted. The conditions can be combined using the <code> </code> character. <b>Default:</b> No.
Class	any string	Valid for all objects except SysNode. Describes the class of the resource object. Used by other programs for various purposes (for example, SNMP agent). <b>Default:</b> <i>type_of_object</i> .
ClusterExclusive	0, 1	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. <b>Default:</b> 0.
Comment	any string	Valid for all objects. No functional meaning in RMS. Used internally by the RMS Wizards. No default value.

Table 8: RMS attributes

Attribute	Possible Values	Description
ControlledShutdown	0, 1	Valid for controlled userApplication objects. If set to 1 for a userApplication, RMS does not send an Offline request to this userApplication, but expects that an explicit request will be generated by a controlling userApplication during its Offline processing. <b>Default:</b> 0.
DetectorStartScript	Any valid detector	Valid for resource object with detector. Specify the detector start command directly in the .us file. No default value.
FaultScript	Valid script (character)	Valid for all object types. Specifies a script to be run if the associated resource enters the Faulted state. No default value.

Table 8: RMS attributes


Attribute	Possible Values	Description
Follow	0, 1	<p>Valid for controller objects. If set to 1, and IgnoreOfflineRequest and IgnoreOnlineRequest are also set to 1, and a parent userApplication object is switched Online, then a controlled userApplication object will also come Online on the same node, regardless of its PriorityList. All requests are propagated only to the controlled userApplication objects running on the same node where the controller is online; the controller keeps track of states of the controlled userApplication objects only on the node where the controller is running.</p> <p>If Follow is 1, then all controlled userApplication objects must be able to run on the same set of nodes as the parent userApplication, although the order of nodes may differ.</p> <p>If Follow is 0, controlled userApplication objects will be switched according to the content of their priority lists. <b>Default:</b> 0.</p> <p> The Follow=0 setting is not currently supported in PRIMECLUSTER version 4.1A10.</p>
Halt	0, 1	<p>Valid for userApplication objects. Eliminates a node if a double fault occurs. <b>Default:</b> 0.</p>

Table 8: RMS attributes

Attribute	Possible Values	Description
HostName	Any SysNode name	Must be set only in the first-level children of a userApplication to assign the particular userApplication to specific nodes and determine the priority of these nodes for the userApplication. No default value.
IgnoreOfflineRequest	0, 1	Valid for controller objects. If set to 1, then neither PreOffline nor Offline requests will be propagated to the controlled userApplication. If 0, then requests will be propagated. <b>Default:</b> 1.
IgnoreOnlineRequest	0, 1	Valid for controller objects. If set to 1, then neither PreOnline nor Online requests will be propagated to the controlled userApplication object. If 0, then Online requests will be propagated. <b>Default:</b> 1.
IgnoreStandbyRequest	0, 1	Valid for controller objects. If set to 1, then neither PreOnline nor Online requests during standby processing will be propagated to the controlled userApplication. If 0, then requests will be propagated. If controller object is not standby capable, then you cannot set IgnoreStandbyRequest to 0. <b>Default:</b> 1.

Table 8: RMS attributes


Attribute	Possible Values	Description
I_List	Any SysNode name	Valid for all SysNode objects. Space-separated list that RMS uses to monitor additional cluster interconnects, which 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. No default value.
IndependentSwitch	0, 1	If 1, then while switching a parent userApplication object from Online to Offline and then to another host, an Online controlled userApplication object will not be switched Offline—it will remain Online on the same host before and after switchover of the parent userApplication object. If 0, then the switch request is propagated from the controlling application to the controlled application by means of the controller object, the application may not be switched on its own. <b>Default:</b> 0.
 The IndependentSwitch=1 setting is not currently supported in PRIMECLUSTER version 4.1A10.		

Table 8: RMS attributes

Attribute	Possible Values	Description
IpAliases	Any SysNode name	Valid for all 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. This setting is restricted to very specific configurations and must never been used in a cluster with CF as interconnect. No default value.
LieOffline	0, 1	Valid for all resource objects. If set to 1, allows the resource to remain Online during Offline processing. <b>Default:</b> 1.
MaxControllers	0 - 512	Valid for userApplication objects. Upper limit of controlling userApplication objects for the specified controlled userApplication. <b>Default:</b> 32000
MonitorOnly	0, 1	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. <b>Default:</b> 0

Table 8: RMS attributes

Attribute	Possible Values	Description
NoDisplay	0, 1	Valid for all object types. If set to 1, specifies that the resource should not be displayed when <code>hvdisp</code> is active. Can be overridden by <code>hvdisp -S</code> . <b>Default:</b> 0
NullDetector	on, off	Valid for resource objects with detector. Used to disable a detector at runtime by setting <code>NullDetector</code> to on. This attribute is for the use with dynamic reconfiguration only. <code>NullDetector</code> must never be set hard-coded to on in the RMS configuration file. <b>Default:</b> off.
OfflineScript	Valid script (character)	Valid for all object types except <code>SysNode</code> objects. Specifies the script to be run to bring the associated resource to the <code>Offline</code> state. No default value.
OfflineDoneScript	Valid script (character)	Valid for <code>userApplication</code> objects. The last script run after the <code>userApplication</code> has completed offline processing. No default value.

Table 8: RMS attributes



Attribute	Possible Values	Description
OnlinePriority	0, 1	<p><b>Valid for userApplication objects. Allows RMS to start the userApplication on the node where it was last Online when the entire cluster was brought down and then restarted. In case of AutoStart or a priority switch, this last-Online node has the highest priority, regardless of its position in the priority list. If set to 1, the userApplication comes Online on the node where it was last Online. If not set (0), the userApplication comes Online on the node with the highest priority in the attribute PriorityList.</b></p> <p><b>RMS keeps track of where the userApplication was last Online by means of timestamps. The node which has the latest timestamp for a userApplication is the node on which the userApplication 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-synchronization between the nodes in the cluster, this responsibility is left to the system administrator.</b></p>

Table 8: RMS attributes

Attribute	Possible Values	Description
		<p>If RMS detects a severe time-discrepancy between the nodes in the cluster, an ERROR message is printed to the switchlog.</p> <p>NTPD may be used to establish consistent time across the machines in the cluster. Please refer to the manual page for <code>xntpd</code> for more information.</p> <p>The <code>OnlinePriority</code> persistent state information will be cleared if the RMS Wizard configuration is activated with the last <code>Online</code> node removed from the configuration. <b>Default:</b> 0.</p>
OnlineScript	Valid script (character)	Valid for all objects except <code>SysNode</code> objects. Specifies the <code>Online</code> script to bring the associated resource to the <code>Online</code> state. No default value.
OnlineTimeout	0 - <i>MAXINT</i>	Valid for <code>controller</code> objects. Specifies the global time in seconds within which a script started by RMS must complete. This global value may be overridden by a node-specific setting of the <code>ScriptTimeout</code> attribute. <b>Default:</b> 0.

Table 8: RMS attributes

Attribute	Possible Values	Description
PersistentFault	0, 1	Valid for userApplication objects. If set to 1, the userApplication keeps track if it was faulted. In case RMS is shutdown and restarted the userApplication returns into the Faulted state, if it was Faulted before, unless the fault is explicitly cleared by either hvutil -c or hvswitch -f or the RMS Wizard configuration is activated with the Faulted SysNode removed from the configuration. <b>Default:</b> 0.
PostOfflineScript	Valid script (character)	Valid for all objects except SysNode objects. Specifies the script to be run after the state of the associated resource changes to Offline. No default value.
PostOnlineScript	Valid script (character)	Valid for all objects except SysNode objects. Specifies the script to be run after the state of the associated resource changes to Online. No default value.
PreCheckScript	Valid script (character)	Valid for userApplication objects. Specifies the script to be forked as the first action during online processing. If the script returns with a zero exit code, online processing proceeds. If the script returns with an exit code other than zero, online processing is not performed and an appropriate warning is logged to the switchlog file. No default value.

Table 8: RMS attributes

Attribute	Possible Values	Description
PreOfflineScript	Valid script (character)	Valid for all objects except <code>SysNode</code> objects. Specifies the script to be run before a resource is taken to the <code>Offline</code> state. No default value.
PreOnlineScript	Valid script (character)	Valid for all objects except <code>SysNode</code> objects. Specifies the script to be run before the associated resource is taken to the <code>Online</code> state. No default value.
PreserveState	0, 1	Valid for <code>userApplication</code> objects. Specifies that resources are not to be taken <code>Offline</code> after a fault. Ignored if <code>AutoSwitchover</code> is set to 1. <b>Default:</b> 0.
PriorityList	Valid name (character)	Valid for <code>userApplication</code> objects. Contains a list of <code>SysNodes</code> where the application can come <code>Online</code> . The order of the list elements determines the next node to which the application is switched during a priority switchover, ordering a switchover after a <code>Fault</code> . RMS uses the order in which nodes are selected and creates <code>PriorityList</code> automatically. You can change the order by selecting individual machines in the list of nodes.

Table 8: RMS attributes

Attribute	Possible Values	Description
Resource	Valid name (character)	Valid for controller objects. One or more names of the controlled userApplication, separated by spaces and/or tabs. The RMS Wizards specify this value in the configuration file. No default value.
rKind	0 - 2047	Specifies the kind of detector for a gResource type object. No default value.
rName	Valid name (character)	Valid for gResource objects. Specifies a string to be forwarded to the generic detector. No default value.
ScriptTimeout	0 - <i>MAXINT</i> (in seconds)	Valid for all object types. Specifies the timeout value for all scripts specified for that node in the configuration file. <b>Default:</b> 300.
ShutdownPriority	0 - <i>MAXINT</i>	Weight factor assigned to userApplication objects, which is used to determine the the priority of the different subclusters in case of interconnect failures and resulting concurrent node elimination requests. The optimal subcluster is defined as the full connected subcluster with the highest weight <i>W</i> . <i>W</i> is the sum of the SF node weight plus the RMS ShutdownPriority of all userApplication objects which are Online in the subcluster. <b>Default:</b> 0

Table 8: RMS attributes

Attribute	Possible Values	Description
SplitRequest	0, 1	Valid for controller objects. If set to 1, then PreOffline and Preonline requests will be propagated to the controlled userApplication separately from the Offline and Online requests. If 0, then separate PreOffline or PreOnline requests will not be issued for the controlled userApplication. Also, if 0, then only Offline and Online requests will be propagated if IgnoreOfflineRequest and IgnoreOnlineRequest respectively are set to 0. <b>Default:</b> 0.
StandbyCapable	0, 1	Valid for resource objects. If set to 1, the object performs Standby Processing on all nodes, where the corresponding userApplication is supposed to be Offline. <b>Default:</b> 0.
StandbyTimeout	0 - <i>MAXINT</i> (in seconds)	Valid for controller objects. The number of seconds to wait before reporting a state change after the controlled userApplication transitions out of Standby state. Default is 0. If attribute Follow is set to 1, StandbyTimeout value must be set to 0. <b>Default:</b> 0.

Table 8: RMS attributes

Attribute	Possible Values	Description
StandbyTransitions	Startup, SwitchRequest, ClearFaultRequest, Valid name (character)	Valid for UserApplication objects. The value specified determines the standby transitions that are to be executed. Startup means that at application startup, the application is requested to go to the Standby state, unless it is already Online or unless it is forced to go Online due to the AutoStartup flag. SwitchRequest means that after switchover, the application that was Online before the switchover will transition to the Standby state. ClearFaultRequest means that the application is requested to go to the Standby state after clearing its state by means of hvutil -c.
WarningScript	Valid script (character)	Valid for resource objects with detector. Specifies the script to be run after the posted state of the associated resource changes to Warning. No default value.

Table 8: RMS attributes





---

## 8 Appendix—Environment variables

You can change the RMS environment by modifying the appropriate entries in the `hvenv.local` file and restarting RMS.



### Caution

RMS environment variables cannot be set in the user environment explicitly. Doing so can cause RMS to lose environment variables settings.



Refer to the Section “Environment variables” for more information concerning environment variables.

### 8.1 Global environment variables

The following list describes the global environment variables for RMS:

#### `HV_AUTOSTARTUP_IGNORE`

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 this 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.

Possible values: List of RMS cluster nodes. The list of RMS cluster nodes must be the names of the `SysNodes` as found in the RMS configuration file. The list of nodes cannot include the CF name. No default value.

#### HV\_AUTOSTART\_WAIT

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.



This attribute is a warning message only. `AutoStartUp` will proceed even if the specified period has expired.

Possible values: 0 - *MAXINT*. The default value is 60 (seconds).

#### HV\_CHECKSUM\_INTERVAL

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.

Possible values: 0 - *MAXINT*. The default value is 120 (seconds).

#### HV\_CM\_EXIT\_WAIT

Interval in seconds that the `hvcn` command waits before exiting after starting up RMS. If RMS fails to start successfully within this period, `hvcn` detects this fault, generates an error message, and terminates with an exit code other than 0.

Possible values: 0 - *MAXINT*. The default value is 2 seconds.

**HV\_LOG\_ACTION\_THRESHOLD**

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” for more information on `HV_LOG_ACTION`).

Possible values: 0 -100. The default value is 98.

**HV\_LOG\_WARN\_THRESHOLD**

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.

Possible values: 0 -100. The default value is 95.

**HV\_RCSTART**

Determines if RMS is started in the `rc` script. (Prerequisite for `rc start: CONFIG.rms` exists and contains a valid entry.)

Possible values: 0, 1. The default value is 1, which starts RMS in the `rc` script.

**RELIANT\_LOG\_LIFE**

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.

Possible values: Any number of days. Default value is 7 (days).

**RELIANT\_LOG\_PATH**

Specifies the directory where all RMS and RMS wizard log files are stored.

Possible values: Any valid path. The default value is `/var/opt/SMAWRrms/log`.

**RELIANT\_PATH**

Specifies the root directory of the RMS directory hierarchy.

Possible values: Any valid path. The default path is `/opt/SMAW/SMAWRrms`. Users do not normally need to change the default setting.

#### RELIANT\_SHUT\_MIN\_WAIT

Defines the period (in seconds) that the command `hvshut` waits before timing out and generating an error message. This value corresponds to the maximum time required by an application to go offline (on all cluster nodes if the `-a hvshut` option is used).

If this value is too low, the `hvshut` command will time out and generate an error message. However, this does not mean that the shutdown process is stopped; it merely means that the `hvshut` command itself will time out. The shutdown process continues within the RMS system. This means that the system shuts down successfully after an `hvshut` command has timed out, even though the command has exited.

Possible values: 0 - *MAXINT*. The default value is 150 (seconds). This variable must be set to the maximum value required to successfully terminate offline processing for a specific application.



The default values of the environment variables are found in `RELIANT_PATH/bin/hvenv`. They can be redefined in the `hvenv.local` command file.

## 8.2 Local environment variables

The following list describes the local environment variables for RMS:

#### HV\_CONNECT\_TIMEOUT

The maximum time (in seconds) that the node detector `hvdet_node` uses for connections to all remote cluster nodes before assuming that the connection attempt has failed.

Possible values: 0 - *MAXINT*. The default value is 0 (seconds). Users do not normally need to change the default setting.

#### HV\_LOG\_ACTION

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`.

Possible values: `on`, `off`. The default value is `off`.

## HV\_MAX\_HVDISP\_FILE\_SIZE

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>`.

Possible values: 0 - *MAXINT*. The default value is 20,000,000 bytes.

## HV\_MAXPROC

Defines the maximum number of scripts RMS can have forked at any time. The default (30) is sufficient in most cases.

Possible values: 0 - *fork limit*. The default value is 30.

## HV\_SYSLOG\_USE

Determines where messages are sent to from the base monitor of RMS.

Possible values: 0, 1. The default setting for this environment variable in `hvenv` is 1. This setting causes RMS to write all RMS ERROR, FATAL ERROR, WARNING, and NOTICE messages to the system log `/var/adm/messages` file. Without any regard of the setting of `HV_SYSLOG_USE`, all correspondent RMS messages also go into the RMS switchlog. However, changing the variable to 0 in `hvenv.local` disables writing to the system log, and all messages will only go to the switchlog.

## RELIANT\_HOSTNAME

The name of the local node in the RMS cluster. The default value of this variable is defined by assigning the output of the CF name with an RMS suffix (for example: `fujii2RMS`) to this variable:

```
export RELIANT_HOSTNAME=`cftool -l 2>/dev/null | tail -1 | cut -f1 -d" " `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 `.us` configuration file. The node name determines IP address RMS uses for establishing contact with this node.

**RELIANT\_INITSRIPT**

Specifies an initialization script to be run by RMS when the system is started. This variable is not set by default. 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, and is not run once for each user application or node.

Possible values: any executable. The default value is `RELIANT_PATH/bin/InitScript`.

**RELIANT\_STARTUP\_PATH**

Defines where RMS searches at start time for the configuration files.

Possible values: any valid path. The default value is `RELIANT_PATH/build`.

**SCRIPTS\_TIME\_OUT**

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.

Possible values: 0 - *MAXINT*. The default value for this environment variable is 300 (seconds).



The default values of the environment variables are found in `RELIANT_PATH/bin/hv env`. They can be redefined in the `hv env.local` command file.

---

## 9 Manual pages

This chapter lists the online manual pages for CCBR, CF, CFS, CIP, Monitoring Agent, PAS, 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
```

### 9.1 CCBR

#### System administration

`cfbackup`

save the cluster configuration information for a PRIMECLUSTER node

`cfrestore`

restore saved cluster configuration formation on a PRIMECLUSTER node

### 9.2 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

`cftool`

print node communications status for a node or the cluster

## 9.3 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

## 9.4 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



## 9.5 Monitoring Agent

### System administration

`clrcimonctl`

Start, stop or restart of the RCI monitoring agent daemon, and display of daemon presence

`clrcumonctl`

Start, stop or restart of the console monitoring agent daemon, and display of daemon presence

## 9.6 PAS

### System administration

`mipcstat`

MIPC statistics

`clmstat`

CLM statistics

## 9.7 RCVM



Applies to transitioning users of existing Fujitsu Siemens products only.

### 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

## 9.8 Resource Database



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

`clstarttrsc`  
resource activation

`clstoptrsc`  
resource deactivation

`clsyncfile`  
distribute a file between cluster nodes

**User command**

`clgettree`  
display the tree information of the resource database

## 9.9 RMS

**System administration**

`hvassert`  
assert (test for) an RMS resource state

`hvcn`  
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

`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

## 9.10 RMS Wizards

**RMS Wizards and RMS Application Wizards**

RMS Wizards are documented as html pages in the SMAWRhvd0 package on the CD-ROM. After installing this package, the documentation is available in the following directory:  
/usr/opt/reliant/htdocs.solaris/wizards.en

## 9.11 SCON

scon  
start the cluster console software

## 9.12 SF

**System administration**

rcsd  
Shutdown Daemon of the Shutdown Facility

rcsd.cfg  
configuration file for the Shutdown Daemon

SA\_rccu.cfg  
configuration file for RCCU Shutdown Agent

SA\_rps.cfg  
configuration file for a Remote Power Switch 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
- sdtool  
interface tool for the Shutdown Daemon

## 9.13 SIS

### System administration

- dtcpadmin  
start the SIS administration utility
- dtcpd  
start the SIS daemon for configuring VIPs
- dtcpstat  
status information about SIS

## 9.14 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

## 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.

## 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 RMS Application Wizards to create object definitions for a specific type of application.

## Application Wizards

See *RMS Application Wizards*.

## 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*.

### **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*.

### **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.



**CF**

See *Cluster Foundation*.

**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*.

**Cluster Foundation**

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 PRIMECLUSTER communications.

**Cluster Join Services (CF)**

This PRIMECLUSTER module handles the forming of a new cluster and the addition of nodes.

**concatenated virtual disk**

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. (Applies to transitioning users of existing Fujitsu Siemens products only.)

See also *mirror virtual disk*, *simple virtual disk*, *striped virtual disk*, *virtual disk*.

**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`.

## Glossary

---

### **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)*.

### **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 (RMS)**

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*.

**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**

This concept applies to the use of redundant resources to avoid single points of failure.

### **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**

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. (Applies to transitioning users of existing Fujitsu Siemens products only.)

See also *mirrored pieces*.

### **mirrored pieces**

Physical pieces that together comprise a mirrored virtual disk. These pieces include mirrored disks and data disks. (Applies to transitioning users of existing Fujitsu Siemens products only.)

See also *mirrored disks*.

### **mirror virtual disk**

Mirror virtual disks consist of two or more physical devices, and all output operations are performed simultaneously on all of the devices. (Applies to transitioning users of existing Fujitsu Siemens products only.)

See also *concatenated virtual disk*, *simple virtual disk*, *striped virtual disk*, *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. (Applies to transitioning users of existing Fujitsu Siemens products only.)

**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.

### **OPS**

See *Oracle Parallel Server*.

### **Oracle Parallel Server**

Oracle Parallel Server allows access to all data in a database to users and applications in a clustered or MPP (massively parallel processing) platform.

### **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)*.

### **primary node (RMS)**

The default node on which a user application comes online when RMS is started. This is always the nodename of the first child listed in the `userApplication` object definition.

### **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*.



**queue**

See *message queue*.

**PRIMECLUSTER services (CF)**

Service modules that provide services and internal interfaces for clustered applications.

**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.

**public LAN**

The local area network (LAN) by which normal users access a machine.

See also *Administrative LAN*.

**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 telecommunications line or LAN.

See also *local node*.

**remote node**

See *remote 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 Application Wizards**

RMS Application Wizards add new menu items to the RMS Wizard Tools for a specific application.

See also *RMS Wizard Tools, Reliant Monitor Services (RMS)*.

**RMS commands**

Commands that enable RMS resources to be administered from the command line.

**RMS configuration**

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 Tools**

A software package composed of various configuration and administration tools used to create and manage applications in an RMS configuration.

See also *RMS Application Wizards, 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)*.

**shared resource**

A resource, such as a disk drive, that is accessible to more than one node.

See also *private resource (RMS)*, *resource (RMS)*.

**simple virtual disk**

Simple virtual disks define either an area within a physical disk partition or an entire partition. (Applies to transitioning users of existing Fujitsu Siemens products only.)

See also *concatenated virtual disk*, *striped virtual disk*, *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**

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). (Applies to transitioning users of existing Fujitsu Siemens products only.)

See also *concatenated virtual disk*, *mirror virtual disk*, *simple virtual disk*, *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 Solaris 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. (Applies to transitioning users of existing Fujitsu Siemens products only.)

See also *concatenated virtual disk*, *mirror virtual disk*, *simple virtual disk*, *striped virtual disk*.

**Web-Based Admin View**

This is a common base to utilize the Graphic User Interface of PRIMECLUSTER. This interface is in Java.

**wizard (RMS)**

An interactive software tool that creates a specific type of application using pretested object definitions. An enabler is a type of wizard.



---

# Abbreviations

**AC**

Access Client

**API**

application program interface

**bm**

base monitor

**CCBR**

Cluster Configuration Backup/Restore

**CF**

Cluster Foundation or Cluster Framework

**CIM**

Cluster Integrity Monitor

**CIP**

Cluster Interconnect Protocol

**CLI**

command-line interface

**CRM**

Cluster Resource Management

**DLPI**

Data Link Provider Interface

**ENS**

Event Notification Services

**GDS**

Global Disk Services

**GFS**

Global File Services

## Abbreviations

---

<b>GLS</b>	Global Link Services
<b>GUI</b>	graphical user interface
<b>HA</b>	high availability
<b>ICF</b>	Internode Communication Facility
<b>I/O</b>	input/output
<b>JOIN</b>	cluster join services module
<b>LAN</b>	local area network
<b>MDS</b>	Meta Data Server
<b>MIB</b>	Management Information Base
<b>NIC</b>	network interface card
<b>NSM</b>	Node State Monitor
<b>OPS</b>	Oracle Parallel Server
<b>OSD</b>	operating system dependant
<b>PAS</b>	Parallel Application Services



<b>RCCU</b>	Remote Console Control Unit
<b>RCI</b>	Remote Cabinet Interface
<b>RMS</b>	Reliant Monitor Services
<b>SA</b>	Shutdown Agent
<b>SAN</b>	Storage Area Network
<b>SCON</b>	single console software
<b>SD</b>	Shutdown Daemon
<b>SF</b>	Shutdown Facility
<b>SIS</b>	Scalable Internet Services
<b>VIP</b>	Virtual Interface Provider



---

# Figures

Figure 1:	Overview of PRIMECLUSTER . . . . .	8
Figure 2:	Relationship between RMS and RMS Wizards . . . . .	12
Figure 3:	NFS Lock Failover screen . . . . .	28
Figure 4:	Main configuration menu when RMS is inactive . . . . .	32
Figure 5:	Application type selection . . . . .	36
Figure 6:	Menu leading to basic settings . . . . .	37
Figure 7:	Menu to configure basic settings . . . . .	38
Figure 8:	Menu to configure non-basic settings . . . . .	39
Figure 9:	Main configuration menu . . . . .	43
Figure 10:	RMS-CreateMachine menu . . . . .	43
Figure 11:	RMS-RemoveMachine menu . . . . .	43
Figure 12:	Main configuration menu . . . . .	44
Figure 13:	Application type selection menu . . . . .	45
Figure 14:	Prompting for further actions . . . . .	46
Figure 15:	Consistency check and Machines+Basics menu . . . . .	47
Figure 16:	List of nodes for failover procedure . . . . .	48
Figure 17:	Machines+Basics menu for additional nodes . . . . .	49
Figure 18:	AutoSwitchOver mode . . . . .	50
Figure 19:	Flags for AutoSwitchOver . . . . .	50
Figure 20:	Saving settings . . . . .	51
Figure 21:	Non-basic settings . . . . .	52
Figure 22:	Prompting for display specification . . . . .	53
Figure 23:	List of display options . . . . .	54
Figure 24:	Successful consistency check for APP1 . . . . .	55

## Figures

---

Figure 25:	Turnkey wizard DEMO . . . . .	56
Figure 26:	Main configuration menu . . . . .	57
Figure 27:	Activating a configuration . . . . .	58
Figure 28:	Quitting the Main configuration menu . . . . .	59
Figure 29:	Starting again with the Main configuration menu . . . . .	60
Figure 30:	Application type selection menu . . . . .	61
Figure 31:	Prompting for further specification . . . . .	62
Figure 32:	Machines+Basics menu . . . . .	63
Figure 33:	List of nodes for failover procedure . . . . .	63
Figure 34:	Machines+Basics menu . . . . .	64
Figure 35:	Non-basic settings . . . . .	65
Figure 36:	Assigning a controller . . . . .	66
Figure 37:	List of applications to be chosen as controlled applications	66
Figure 38:	Menu for setting controller flags . . . . .	67
Figure 39:	Controller timeout period . . . . .	67
Figure 40:	Changing controller timeout period . . . . .	68
Figure 41:	Saving flags for controller . . . . .	68
Figure 42:	Indication of flags set for controller . . . . .	69
Figure 43:	Menu with settings for GENERIC turnkey wizard . . . . .	70
Figure 44:	Main configuration menu . . . . .	71
Figure 45:	Main configuration menu . . . . .	72
Figure 46:	Activating the configuration for the second time . . . . .	73
Figure 47:	Return to Main configuration menu . . . . .	74
Figure 48:	Invoking the Cluster Admin GUI . . . . .	78
Figure 49:	Web-Based Admin View login screen . . . . .	79
Figure 50:	Top menu . . . . .	80

---

Figure 51:	Cluster menu . . . . .	81
Figure 52:	Main screen . . . . .	82
Figure 53:	RMS main window . . . . .	83
Figure 54:	RMS tree with a controller object . . . . .	84
Figure 55:	Configuration information or object attributes . . . . .	85
Figure 56:	Command pop-up . . . . .	86
Figure 57:	Command pop-up for an Offline application . . . . .	87
Figure 58:	Viewing the RMS switchlog file . . . . .	88
Figure 59:	Viewing the RMS switchlog file in a detached window . . . . .	89
Figure 60:	Viewing the application log . . . . .	90
Figure 61:	Search based on date and time filter . . . . .	91
Figure 62:	Using the Find pop-up in log viewer . . . . .	92
Figure 63:	RMS full graph . . . . .	94
Figure 64:	RMS application graph . . . . .	95
Figure 65:	RMS sub-application graph . . . . .	96
Figure 66:	Composite sub-application graph . . . . .	97
Figure 67:	Configuration information pop-up . . . . .	98
Figure 68:	Command pop-up . . . . .	99
Figure 69:	Graph with affiliations . . . . .	100
Figure 70:	Graphs with affiliations and names . . . . .	101
Figure 71:	RMS graph customization . . . . .	102
Figure 72:	RMS graph after RMS is shut down . . . . .	103
Figure 73:	Clusterwide table . . . . .	104
Figure 74:	Primary node shown in clusterwide table . . . . .	104
Figure 75:	Faulted and Offline applications in the clusterwide table . . . . .	105
Figure 76:	Exclamation marks in clusterwide table and the RMS tree . . . . .	106

## Figures

---

Figure 77:	Command pop-ups in clusterwide table . . . . .	107
Figure 78:	Before RMS is shut down . . . . .	108
Figure 79:	After RMS is restarted with a different configuration . . . . .	109
Figure 80:	Starting RMS . . . . .	111
Figure 81:	RMS Start Menu for all nodes . . . . .	112
Figure 82:	RMS Start Menu for individual nodes . . . . .	113
Figure 83:	Starting RMS on individual nodes . . . . .	114
Figure 84:	Stopping RMS . . . . .	115
Figure 85:	Stopping RMS on all available nodes . . . . .	116
Figure 86:	Stopping RMS on one node from the list . . . . .	117
Figure 87:	Using command pop-up to stop RMS . . . . .	118
Figure 88:	Starting an application . . . . .	119
Figure 89:	Switching an application . . . . .	121
Figure 90:	Switching a busy application . . . . .	122
Figure 91:	Shutting down an application . . . . .	124
Figure 92:	Clearing an application fault . . . . .	125
Figure 93:	Clusterwide environment variables . . . . .	127
Figure 94:	Local environmental variables pop-up . . . . .	128
Figure 95:	Local environmental variables window . . . . .	129
Figure 96:	Displaying application states . . . . .	130
Figure 97:	Invoking the log viewer . . . . .	139
Figure 98:	Find pop-up window . . . . .	140
Figure 99:	Detached log . . . . .	141
Figure 100:	Resource-based search . . . . .	142
Figure 101:	Results of time-based search . . . . .	143

Figure 102: Results of keyword-based search . . . . . 144

Figure 103: Results of severity-level-based search . . . . . 146





---

# Tables

Table 1: Available CLI commands . . . . .	17
Table 2: RMS base directory structure . . . . .	22
Table 3: Log directory structure . . . . .	24
Table 4: Log files . . . . .	137
Table 5: Descriptions of severity levels . . . . .	145
Table 6: Log levels . . . . .	147
Table 7: Object types . . . . .	277
Table 8: RMS attributes . . . . .	279



---

# Index

## A

- activating
  - application 123
  - configuration 34, 57
  - configuration second time 72
- additional error messages 252
- administrative privileges 79
- Affiliation, attribute 279
- andOp
  - attributes 277
  - description 277
- application graph 95
- application log files 88
- application logs
  - displaying 90
  - searching text 92
- Application Wizards 11, 13
  - configuration 26
  - overview 8
- applications
  - activating 123
  - dependencies 84
  - displaying states 130
  - taking offline 124
  - viewing logs 131
- attributes
  - Affiliation 279
  - AutoRecover 279
  - AutoRecoverCleanup 279
  - AutoStartUp 280
  - AutoSwitchOver 280
  - Class 280
  - ClusterExclusive 280
  - Comment 280
  - ControlledShutdown 281
  - DetectorStartScript 281
  - FaultScript 281
  - Follow 282
  - Halt 282
  - HostName 283
  - I\_List 284
  - IgnoreOfflineRequest 283
  - IgnoreOnlineRequest 283
  - IgnoreStandbyRequest 283
  - IndependantSwitch 284
  - IpAliases 285
  - LieOffline 285
  - MaxControllers 285
  - MonitorOnly 285
  - NoDisplay 286
  - NullDetector 286
  - OfflineDoneScript 286
  - OfflineScript 286
  - OnlinePriority 287
  - OnlineScript 288
  - OnlineTimeout 288
  - PersistentFault 289
  - PostOfflineScript 289
  - PostOnlineScript 289
  - PreCheckScript 289
  - PreOfflineScript 290
  - PreOnlineScript 290
  - PreserveState 290
  - PriorityList 290
  - Resource 291
  - rName 291
  - ScriptTimeout 291
  - ShutdownPriority 291
  - SplitRequest 292
  - StandbyCapable 292
  - StandbyTimeout 292
  - WarningScript 293
- AutoRecover, attribute 279
- AutoRecoverCleanup, attribute 279
- AutoStartUp, attribute 280
- AutoSwitchOver, attribute 280

## B

- base monitor 14
  - debug messages 135
  - detectors 41
  - error code 155

- high availability 8
- log file 137
- log level 147
- log levels 147
- messages 137
- stack tracing 147
- switchlog 150
- basic settings, wizards 37
- bin, directory 22
- bmlog description 137
- browser 77
- build, directory 22
- C**
- CF commands
  - cfconfig 301
  - cftool 301
- cfset 301
- cfsmntd 302
- CIP commands
  - cip.cf 302
  - cipconfig 302
  - ciptool 302
- Class, attribute 280
- clbackuprdb 304
- clearing
  - faulted resources 19
  - Faulted state 125
  - faults 125
  - hung nodes 19
  - SysNode Wait state 126
- clgettree 305
- CLI
  - See* RMS CLI
- clinitreset 304
- clrestorerdb 304
- clsetparam 304
- clsetup 304
- clstartsrc 304
- clstoprsc 304
- cluster 1
  - high availability 8
  - services 7
  - switching user applications 17
- Cluster Admin 14
  - administrative privileges 79
  - application graph 95
  - application log files 88
  - clusterwide table 104
  - command pop-ups 86
  - configuration 98
    - graph 97
    - GUI 77
    - logging in 79
    - main screen 81
    - object attributes 85
    - operator privileges 79
    - overview 8
    - primary management server 78
    - procedures 110
    - RMS graphs 93
    - RMS main window 83
    - RMS tree 83
    - root privileges 79
    - searching log text 92
    - secondary management server 78
    - starting 77
    - switchlog 88
    - switchlog panel 89
    - SysNode selection 86
    - userApplication selection 86
    - using 77
  - cluster file system 7
  - cluster node
    - detector timeout for remote 298
    - ignore at startup 295
    - wait to report online 296
  - cluster volume management 7
  - ClusterExclusive, attribute 280
  - clusterwide table 104
  - Cmdline, resource wizard 27
  - command pop-ups
    - clusterwide table 107
    - RMS graph 99
    - RMS tree 86
  - Comment, attribute 280
  - composite sub-application graph 97

- configuration information, graphs 98
- configurations
  - creating 30
  - displaying 17
  - displaying information 82, 83
  - editing 30
  - graph 93
  - individual node details 85
  - wizard sample 42
- configuring
  - applications 25, 27
  - disk class 27
  - disk groups 27
  - file systems 27
  - IP addresses 27
  - resources 27
- ControlledShutdown, attribute 281
- controller
  - attributes 277
  - dependencies 84
  - description 277
  - graph 97
- Controller, resource wizard 27
- creating
  - application 42
  - configuration 30
  - second application 59
- D**
- debug level, wizards 154
- debug messages 133
  - base monitor 135
  - log directory 135
  - severity level 145
  - wizards 151, 153
- debug reporting, wizards 154
- DEBUG statements, wizards 154
- defining timeout 298
- DEMO turnkey wizard 31, 42, 45
- dependant resources 8
- dependencies 84
- det\_disklog file 138
- detectors 8
  - RMS Application Wizards 13
  - RMS Wizard Tools 13
  - starting 114
- DetectorStartScript, attribute 281
- directories
  - bin 22
  - build 22
  - etc 22
  - include 22
  - lib 23
  - tab 23
  - us 23
- directory hierarchy
  - root directory 297
  - specifying root directory 297
- disk classes 10
- displaying
  - application states 130
  - current RMS configuration 17
  - environment variables 127
- dkconfig 303
- dkmigrate 303
- dkmirror 303
- dktab 304
- documentation
  - additional 2
  - wizards 75
- E**
- ENV
  - attributes 277
  - description 277
- environment variables 20
  - displaying 127
  - HV\_AUTOSTART\_WAIT 296
  - HV\_AUTOSTARTUP\_IGNORE 295
  - HV\_CHECKSUM\_INTERVAL 296
  - HV\_CONNECT\_TIMEOUT 298
  - HV\_MAXPROC 299
  - RELIANT\_HOSTNAME 299
  - RELIANT\_INITSCRIPT 300
  - RELIANT\_LOG\_LIFE 297
  - RELIANT\_LOG\_PATH 135, 297

- RELIANT\_PATH 297
- RELIANT\_SHUT\_MIN\_WAIT 298
- SCRIPTS\_TIME\_OUT 300
- ENVL
  - attributes 277
  - description 277
- error messages 133, 156
  - additional 252
  - base monitor 135
  - fatal 241
- etc, directory 22
- F**
- fatal error messages 241
- Faulted state 15
  - clearing 125
  - FaultScript 16
- faults
  - clearing 125
  - failover 45
- FaultScript
  - attribute 281
  - script 16
- file systems
  - defining resources 10
  - filling up 154
  - Fsystem 27
  - resource type 13
  - resources 25
  - warning threshold 297
- fjsvwwbs 307
- fjsvwwcnf 307
- Follow, attribute 282
- forced shutdown 117
- fsck\_rcfs 302
- Fsystem, resource wizard 27
- G**
- GDS
  - See* Global Disk Services
- Gds, resource wizard 27
- Global Disk Services, configuring disk class 27
- graphical user interface
  - See* GUI
- graphs
  - application 95
  - command pop-ups 99
  - composite sub-application 97
  - configuration information 98
- gResource
  - object type 277
  - required attributes 277
- GUI
  - after shut down 103
  - messages 131
  - pull-down menus 82
  - starting RMS 83
- H**
- Halt, attribute 282
- high availability 1, 7
  - configuring for 41
  - specifying applications 33
  - wizards 26
- HostName, attribute 283
- HV\_AUTOSTART\_WAIT 296
- HV\_AUTOSTARTUP\_IGNORE 295
- HV\_CHECKSUM\_INTERVAL 296
- HV\_CONNECT\_TIMEOUT 298
- HV\_MAXPROC 299
- hvcn -l 147
- hvdsp 130
  - file size 299
  - no display 286
- hvenv
  - changing variables 21
  - values 20
- hvexec 41
- hvshut
  - defined 18
  - defining timeout 298
  - stopping RMS 118
- hvswitch
  - defined 18
  - userApplication 120

- hvutil
  - defined 19
  - shutting down an application 124
- hvw command 13
  - defined 31
  - editing configurations 42
  - operation mode 35
  - resuming configuration 60
- I**
- I\_List, attribute 284
- IgnoreOfflineRequest, attribute 283
- IgnoreOnlineRequest, attribute 283
- IgnoreStandbyRequest, attribute 283
- include, directory 22
- IndependantSwitch, attribute 284
- initialization script, specifying 300
- InitScript 16
- IP addresses
  - defining resources 10
  - resource wizard 27
- Ipaddress, resource wizard 27
- IpAliases, attribute 285
- L**
- LAN interfaces 30
- LEFTCLUSTER 312, 315, 317
- lib, directory 23
- LieOffline, attribute 285
- log files
  - application 88
  - base monitor 137
  - interpreting 149
  - node detector 138
  - specify directory 297
  - switchlog 138
  - time of preservation 297
  - viewing 131
- log levels, specifying 147
- log messages, wizards 151
- logging in, Cluster Admin 79
- M**
- MA commands
  - clrcumonctl 303
  - clrcimonctl 303
- main menu
  - Cluster Admin 111
  - wizards 31, 32
- management server 78
- manual pages
  - display 301
  - listing 301
- MaxControllers, attribute 285
- messages
  - additional 252
  - base monitor 137
  - bmlog 137
  - debug 135, 145
  - error 135, 156
  - fatal 241
  - generic detector log 138
  - node detector 138
  - troubleshooting RMS 274
  - wizards 151
- mirror virtual disks 316
- monitoring states 9
- MonitorOnly, attribute 285
- mount\_rcfs 302
- N**
- ngadmin 302
- node detector log files 138
- NoDisplay, attribute 286
- non-basic settings, wizards 37
- NullDetector, attribute 286
- O**
- object types
  - andOp 277
  - controller 277
  - ENV 277
  - ENVL 277
  - gResource 277
  - orOp 278

- SysNode 278
  - userApplication 278
  - objects
    - activating applications 123
    - attributes 20, 85
    - clearing a fault 125
    - clusterwide table 104
    - command pop-ups 99, 107
    - controller 84, 97
    - dependencies 93
    - graph customization 100
    - information 98
    - relationships 93
    - RMS full graph 93
    - RMS tree 86
    - selecting 86
    - starting an application 119
    - switching applications 121
    - SysNode 86
    - taking application offline 124
    - userApplication 86, 118
  - offline processing 9
  - offline scripts 13
  - Offline state 15
    - OfflineDoneScript 16
    - OfflineScript 16
    - PostOfflineScript 16
  - OfflineDoneScript
    - attribute 286
    - script 16
  - OfflineFault state 15
  - OfflineScript 16
  - OfflineScript, attribute 286
  - online processing 9
  - online scripts 13
  - Online state 15
    - OnlineScript 16
    - PostOnlineScript 16
    - PreCheckScript 16
    - PreOnlineScript 16
  - OnlinePriority, attribute 287
  - OnlineScript
    - attribute 288
    - script 16
  - OnlineTimeout, attribute 288
  - operator privileges 79
  - orOp, object type 278
- ## P
- parallel application support 7
  - PAS commands
    - clmtest 303
    - mipcstat 303
  - PersistentFault, attribute 289
  - PostOfflineScript
    - attribute 289
    - script 16
  - PostOnlineScript
    - attribute 289
    - script 16
  - PreCheckScript
    - attribute 289
    - script 16
  - PreOfflineScript
    - attribute 290
  - PreOnlineScript
    - attribute 290
    - script 16
  - PreserveState, attribute 290
  - primary management server 78
  - PRIMECLUSTER 7
  - PriorityList, attribute 290
  - privileges 79
  - procedures 110
  - pseudo device driver 323
- ## R
- RAID 322
  - rcfs\_fumount 302
  - rcfs\_list 302
  - rcfs\_switch 302
  - rcsd 306
  - rcsd.cfg 306
  - Reliant Monitor Services
    - clusterwide table 104
    - components 14
    - full graph 93
    - graphs 93



- high availability 8
  - main window 83
  - overview 8
  - tree 83
- RELIANT\_HOSTNAME 299
- RELIANT\_INITSCRIPT 300
- RELIANT\_LOG\_LIFE 297
- RELIANT\_LOG\_PATH 135, 297
- RELIANT\_PATH 297
- RELIANT\_SHUT\_MIN\_WAIT 298
- request-triggered scripts
  - InitScript 16
  - OfflineScript 16
  - OnlineScript 16
  - PreCheckScript 16
  - PreOnlineScript 16
- resource wizards
  - Cmdline 27
  - Controller 27
  - Fsystem 27
  - Gds 27
  - Ipaddress 27
  - Rcvm 27
  - Vxvm 27
- Resource, attribute 291
- resource-oriented wizards 27
- resources
  - clearing faulted 19
  - configuring 27
  - defining 10
  - dependant 8
  - executing scripts 35
  - file system entries 29
  - LAN interfaces 30
  - monitoring 41
  - non-basic settings 39
  - object types 19
  - scripts 15
  - shared remote entries 29
  - site preparation 28
  - states 8
  - wizards 27
- rKind 291
- RMS
  - See* Reliant Monitor Services
- RMS Application Wizards 11, 13
  - detectors 13
  - hvw command 13
  - overview 8
  - scripts 13
- RMS CLI 17
  - hvassert 17
  - hvcn 17
  - hvconfig 17
  - hvdsp 17
  - hvdist 18
  - hvdump 18
  - hvgdmake 18
  - hvlogclean 18
  - hvshut 18
  - hvswitch 18
  - hvthrottle 19
  - hvutil 19
  - options 110
  - switching userApplication 17
- RMS commands
  - hvassert 305
  - hvcn 17, 114, 305
  - hvconfig 17, 305
  - hvdsp 17, 130, 305
  - hvdist 18, 305
  - hvdump 18, 305
  - hvenv.local 306
  - hvexec 41
  - hvgdmake 18, 305
  - hvlogclean 18, 305
  - hvshut 18, 118, 305
  - hvswitch 18, 120, 305
  - hvthrottle 19, 305
  - hvutil 19, 124, 305
- RMS Wizard Tools 10
  - detectors 13
  - hvw command 13
  - overview 8
  - resource types 13
  - scripts 13

## Index

---

- RMS wizards
  - See* wizards 42
- rName, attribute 291
- root privileges 79
- running processes 10
  
- S**
- SA\_rccu.cfg 306
- SA\_rps.cfg 306
- SA\_scon.cfg 306
- SA\_sspint.cfg 307
- SA\_sunF.cfg 307
- SA\_wtinps.cfg 307
- sample configuration, wizards 42
- scalability 7
- scon 306
- scripts 8, 16
  - Offline 13
  - Online 13
  - resources 15
  - RMS Application Wizards 13
  - timeout 300
- SCRIPTS\_TIME\_OUT 300
- ScriptTimeout, attribute 291
- sdtool 307
- searching log text 92
- secondary management server 78
- secondary menus, wizards 36
- severity levels
  - Alert 145
  - Critical 145
  - Debug 145
  - Emergency 145
  - Error 145
  - Info 145
  - Notice 145
  - Warning 145
- Shutdown Facility 9
- ShutdownPriority, attribute 291
- simple virtual disks 321
- SIS commands
  - dtcpadmin 307
  - dtcpd 307
  - dtcpdbg 307
  - software monitor
    - function 1
    - RMS 8
  - SplitRequest, attribute 292
  - Standby state 15
  - StandbyCapable, attribute 292
  - StandbyTimeout, attribute 292
  - starting
    - an application 119
    - RMS 111
  - startup file 114
  - states 15
    - displaying information 104
    - Faulted 15
    - Offline 15
    - OfflineFault 15
    - Online 15
    - Standby 15
    - Wait 126
    - Warning 15
  - state-triggered scripts
    - FaultScript 16
    - OfflineDoneScript 16
    - PostOfflineScript 16
    - PostOnlineScript 16
    - WarningScript 16
  - stopping RMS 115
  - sub-application graph 96
  - sub-applications 84
  - sub-menus, wizards 36
  - summary table 104
  - switching an application 121
  - switchlog
    - error messages 156
    - file 135, 138
    - panel 89
    - viewing 88, 131
  - SysNode
    - clearing Wait state 126
    - object selection 86
    - object type 278
  - SysNode Wait state, clearing 126

**T**

- tab, directory 23
- tables
  - clusterwide 104
  - command pop-ups 107
- taking an application offline 124
- turning off wizard debug output 153
- turnkey wizards
  - DEMO 31, 42
  - See also* wizards 42

**U**

- us, directory 23
- userApplication
  - activating 123
  - clearing fault 125
  - hvswitch 120
  - object 9
  - object selection 86
  - object type 278
  - RMS tree 84
  - state information 104
  - taking Offline 124
  - with hvshut 118

**V**

- vdisk 303
- viewing
  - application logs 131
  - GUI message 131
  - switchlogs 131
- virtual disks 1
  - mirror 316
  - simple 321

**W**

- Wait state
  - clearing faulted resources 19
  - clearing hung nodes 19
  - clearing SysNode 126
- Warning state 15
  - WarningScript 16
- WarningScript 16
- WarningScript, attribute 293

**Web-Based Admin View**

- login 79
- primary management server 78
- secondary management server 78

**Wizard Tools 26**

- wizards
  - basic settings 37
  - configuring 10, 26
  - creating a configuration 30
  - debug level 154
  - debug messages 153
  - debug reporting 154
  - DEBUG statements 154
  - DEMO turnkey 31
  - frequently used items 39
  - general description 26
  - hvxexec command 41
  - main menu 31, 32
  - non-basic settings 37
  - resource wizards 27
  - sample configuration 42
  - secondary menus 36
  - sub-menus 36
  - turn off debug output 153
- wizards log messages 151
- wwCntl 307
- wwGetparam 307
- wwSetparam 308
- wwstat 308



Fujitsu Siemens Computers GmbH  
User Documentation  
33094 Paderborn  
Germany

Comments  
Suggestions  
Corrections

**Fax: (++49) 700 / 372 00001**

email: [manuals@fujitsu-siemens.com](mailto:manuals@fujitsu-siemens.com)  
<http://manuals.fujitsu-siemens.com>

---

Submitted by

---

Comments on PRIMECLUSTER™  
Reliant Monitor Services (RMS)  
Configuration and Administration Guide (Solaris)





Fujitsu Siemens Computers GmbH  
User Documentation  
33094 Paderborn  
Germany

Comments  
Suggestions  
Corrections

**Fax: (++49) 700 / 372 00001**

email: [manuals@fujitsu-siemens.com](mailto:manuals@fujitsu-siemens.com)  
<http://manuals.fujitsu-siemens.com>

---

Submitted by

---

Comments on PRIMECLUSTER™  
Reliant Monitor Services (RMS)  
Configuration and Administration Guide (Solaris)



