Using the Event Monitoring Service

Manufacturing Part Number: B7612-90015

November 1999

Legal Notices

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Corporate Offices:

Hewlett-Packard Co.
3000 Hanover St.
Palo Alto, CA 94304

Use, duplication or disclosure by the U.S. Government Department of Defense is subject to restrictions as set forth in paragraph (b)(3)(ii) of the Rights in Technical Data and Software clause in FAR 52.227-7013.

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Use of this manual and flexible disc(s), compact disc(s), or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Contents

Contents

Contents

Contents

Printing History

Table 1

Printing Date	Part Number	Edition
March 1999	B7612-90009	Edition 1
November 1999	B7612-90015	Edition 2

This edition documents material related to using the Event Monitoring Service to create monitoring requests for system resources.

The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The part number is revised when extensive technical changes are incorporated.

New editions of this manual will incorporate all material updated since the previous edition.

HP Printing Division:

Business Critical Computing Business Unit (BCC)
Hewlett-Packard Co.
19111 Pruneridge Ave.
Cupertino, CA 95014

Preface

This guide describes how to use the Event Monitoring Service (EMS) and how to configure Management Information Base (MIB) monitors. The MIB monitors check and report status on cluster, network, and system resources.

EMS functions at various levels:

- independently
- with high availability software such as ServiceGuard
- with enterprise management products such as IT/O

The contents of this guide are as follows:

- Understanding the Event Monitoring Service
- Selecting Resources to Monitor
- Defining a Monitoring Request
- Changing Monitoring Requests
- Monitoring ServiceGuard Package Dependencies
- Monitoring Cluster Resources
- Monitoring Network Interfaces
- Monitoring System Resources
- Dictionary File Command Line Options
- Troubleshooting

Related Publications

The following documents contain additional related information:

- *Using High Availability Monitors* (HP Part Number B5736-90025)
- *EMS Hardware Monitors User's Guide* (HP Part Number B6191-90018)
- *Managing MC/ServiceGuard* (HP Part Number B3936-90026)
- *Configuring OPS Clusters with ServiceGuard OPS Edition* (HP Part Number 5158-90026)
- *Managing Systems and Workgroups* (HP Part Number B2355-90664)

- Peter Weygant, *Clusters for High Availability: A Primer of HP-UX Solutions* (ISBN 0-13-494758-4). HP Press: Prentice Hall, Inc., 1996
- Tom Madell, *Disk and File Management Tasks on HP-UX* (ISBN 0-13-518861-X). HP Press; Prentice Hall, Inc., 1997
- *HP OpenView IT/Operations Admnistrator's Reference* (HP Part Number B6941-90001)
- *Managing Highly Available NFS* (HP Part Number B5125-90001)
- **`http://docs.hp.com`** Web site for information about Hewlett-Packard's high-availability technologies where you can find documents.Select HP-UX then `High Availability`.
- **`www.software.hp.com`** Web site for designing and building an EMS monitor. Select High Availability then Event Monitoring Service Developer's Kit.

Problem Reporting

If you have any problems with the software or documentation, please contact your local Hewlett-Packard Sales Office or Customer Service Center.

1 Understanding the Event Monitoring Service

The Event Monitoring Service (EMS) is a framework for resource monitoring. Use EMS to monitor system resources including configuring, checking resource status, and sending notification when configured conditions are met.

This chapter describes the following:

- Event Monitoring Service Overview
- EMS Requirements
- EMS Resource Classes
- Client and Target Applications
- Resource Monitors
- EMS Framework Components

Event Monitoring Service Overview

The Event Monitoring Service (EMS) monitors system resources. Use EMS to configure monitoring requests, check resource status, and send notification when configured conditions are met.

EMS can work in a high availability environment. It can report a loss of redundant resources. Identifying and reporting single points of failure helps maintain a proactive approach to preventing the loss of data and availability.

EMS only observes a system, and does not modify the system. Use EMS with additional software to take or specify action.

The three basic components of EMS are:

- Client and Target Applications

  System administrators use client applications to set, modify, or remove monitoring requests. System administrators use target applications to receive event notifications and possibly take actions.

  Client applications include ServiceGuard (MC/ServiceGuard or ServiceGuard OPS Edition), the Event Monitoring Service (EMS) GUI (Graphical User Interface), or other applications that complies with the EMS API.

  The target application can be any application that supports the EMS protocols. The supported protocols are:

  — TCP/IP or UDP/IP

    This includes any application that accepts these protocols and follows the rules defined in the EMS Developer's Kit.

  — `opcmsg` method (for ITO)

    This option is used for IT/Operations notifications.

  — SNMP traps

    This option can be used with any application that accepts SNMP traps, such as NNM or IT/O. You need to set up the application to recognize the SNMP traps generated.

  — email

This option does not require any extra handling. Specify the email address when the monitoring request is created.

— syslog and textlog

This option does not require any extra handling. Specify the log file when the monitoring request is created. Syslog notifications go to the local system.

— console

This option does not require any extra handling. Specify the console when the monitoring request is created. Notifications go to the local system.

— ServiceGuard

This option requires that the client and target application both be ServiceGuard running on the same local system.

- Resource Monitors

  Resource monitors observe designated resources and report back resource values or events to the Event Monitoring Service.

  Hewlett-Packard provides monitors with the High Availability Monitors package and with the Event Monitoring Service. The monitors available through Hewlett-Packard include: HA Database Monitor, HA Disk Monitor, HA Cluster Monitor, HA Network Interface Monitor, and HA System Resource Monitor.

- Event Monitoring Service Framework

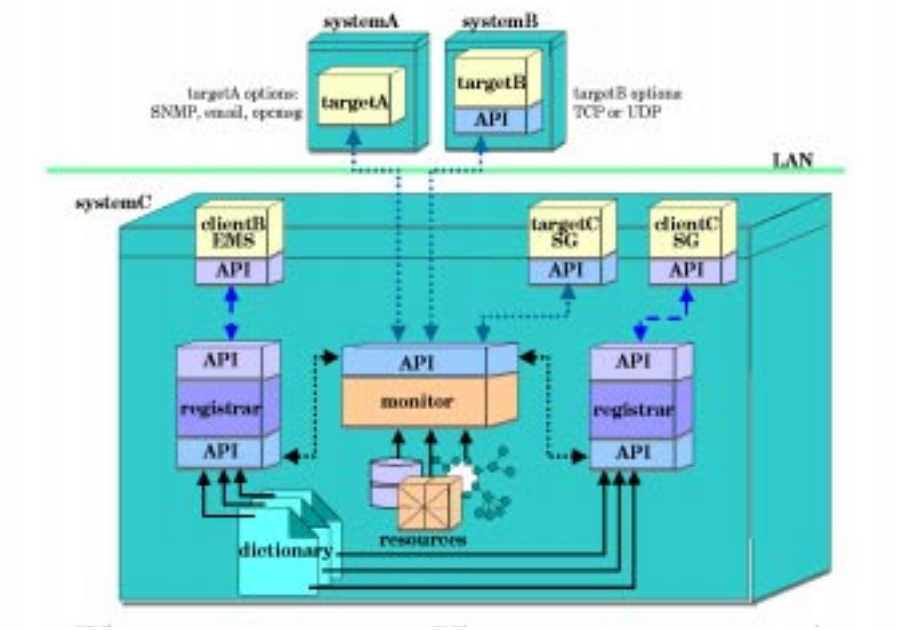
  The EMS Framework provides the interface between the client applications, monitors, and target applications.

  The EMS Framework contains the Applications Programmers Interface (API), registrar, and the Resource Dictionary.

  Developers use the API to create additional monitors for use with client and target applications, such as the EMS GUI or ServiceGuard. Monitor components to be created include: resource dictionary, resource monitor binary file, man page (recommended), message catalog (recommended)

Figure 1-1 shows the relationships between the Event Monitoring Service components.

Figure 1-1 Event Monitoring Service Components



The process is as follows:

1. The system administrator enters the *client application*, for example, the EMS GUI or `High Availability Clusters` area of SAM, to begin the discovery phase of creating a monitoring request.

   The discovery phase, includes identifying the resources to be monitored and configuring the request. It can be accomplished through many methods, including:

   - EMS GUI
   - ServiceGuard
   - monconfig utility
   - resls or resdata commands

2. The *EMS API* provides the interface between the client request and the registrar. There is a one to one correspondence between the client and registrar.
3. The *registrar* refers to the dictionary for a list of available resources and related monitors.

The resources listed in the dictionary are passed back to the client.

4. When a discovery request is made that exceeds the scope of the information in the dictionary, the *registrar* launches the appropriate resource monitor application, if it is not already running, and passes the request on to the monitor. Multiple registrars may access the same monitor.
5. The *EMS API* provides the interface between the registrar and the monitor.
6. The *monitor* identifies the resources. The list of resources is passed back through the registrar to the client requestor.
7. The system administrator, through the *client* application:
   - continues to drill down through the list of available resources supplied by the registrar, dictionary, and monitor
   - identifies the resources to monitor
   - completes the monitoring request defines conditions of where and how to send event notification

   A completed monitoring request identifies:
   - what resources to monitor
   - what events to watch for and how often
   - what notifications to send when an event occurs
   - where to send notifications

   Events are defined for either of two resource state types:
   - periodic checking against either thresholds or state/value changes
   - continuous checking for asynchronous stateless events
8. The *registrar* passes completed monitoring requests down to the appropriate resource monitor application.
9. The *monitor* checks the resource as specified in the monitor request. It passes back to the EMS API whether the request is accepted or rejected and as appropriate, why a request is rejected.
10. The EMS API provides the interface between the monitor and the target.
11. The monitor begins collecting data as specified in the monitoring

request.

12. The *EMS API* interprets the information received from the monitor, determines if an event occurred, and forwards the notification to the target applications. The method of informing the target application of a critical resource value can vary for different target applications.

    In the case of ServiceGuard, the client application and the target application are the same and reside on the same system.

EMS Requirements

The following are system requirements for the Event Monitoring Service:

- All hardware you intend to monitor, such as disks and LAN cards, have been configured and tested prior to configuring EMS.
- EMS must be installed on an HP 9000 Series 700 or Series 800 system running HP-UX version 10.20 or later.

When installing one or more EMS components, check that the version levels for the other components are compatible. Refer to the recent Release Notes for each component, such as EMS, HA Monitors, or ServiceGuard.

1. Go to the web site:

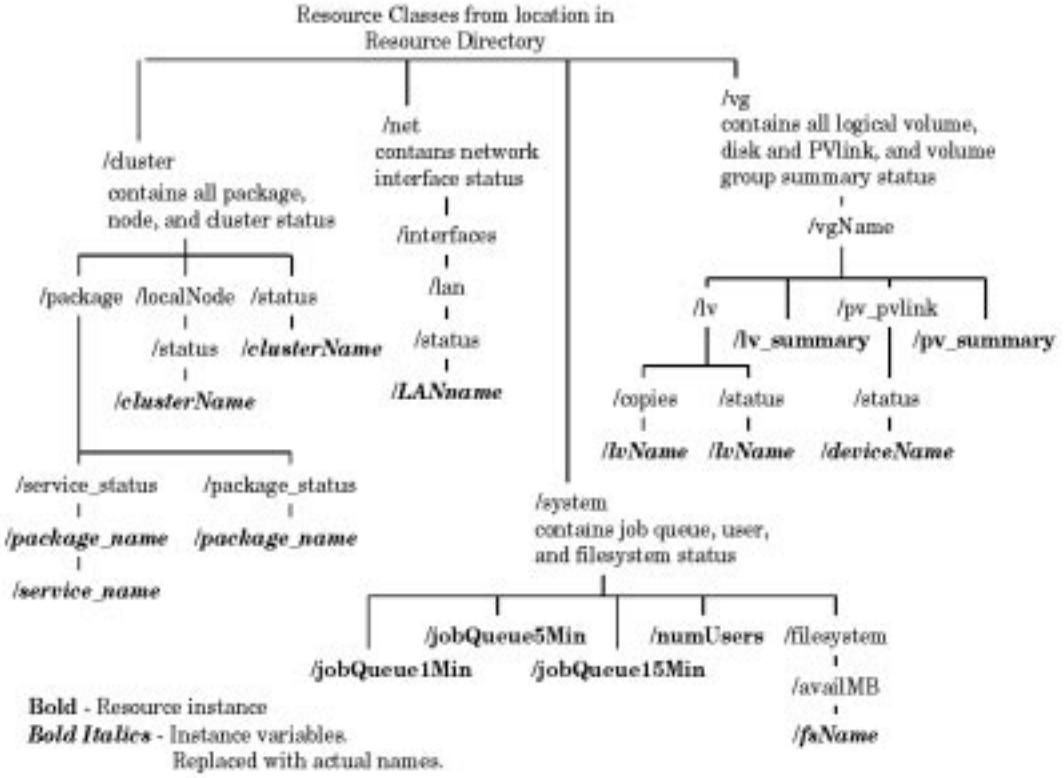
   `http://docs.hp.com`.

2. From the web site select HP-UX, then `High Availability`, then scroll to identify and select the latest Release Notes for each component.

EMS Resource Classes

EMS groups resources into classes in a hierarchy similar to that of a filesystem structure. Figure 1-2 is a example of a resource hierarchy.

Figure 1-2 Event Monitoring Service Resource Class Hierarchy



Client and Target Applications

This section describes some of the client and target application options and processes. Target applications can be written using the EMS API.

EMS with ServiceGuard

ServiceGuard can be configured with EMS to monitor the health of selected resources, such as disks. Based on the status of the resources, ServiceGuard can decide to fail packages over. When working with EMS, ServiceGuard acts as both the client and target application. EMS works with both the MC/ServiceGuard and ServiceGuard OPS Edition.

Configure EMS requests for use with ServiceGuard packages either:

- through the Package Configuration in the Cluster area of SAM
- by editing the package configuration ASCII file

In addition, is it recommended that you also create requests through EMS to:

- enable a redundant notification system
- monitor events that affect high availability
- be alerted to the cause of a package failover

ServiceGuard may already be configured to monitor the health of nodes, services, and subnets, and to make failover decisions based on the status of these resources. Using EMS with ServiceGuard adds to the set of failures or events that trigger failover and affect availability.

EMS GUI Client Application

Use the EMS GUI, found in the Resource Management area of SAM, to create monitoring requests for resources and targets. The EMS GUI starts from the graphical version of SAM. Click through and select from the various screens to define your monitoring request. The options include:

1. Select resources to be monitored.

   The full path of a resource includes the resource class hierarchy and

instance. An example of a full resource path for the physical volume status of the device `/dev/dsk/c0t1d2` belonging to volume group `vgDataBase`, is `/vg/vgDataBase/pv_pvlink/status/c0t1d2`.

2. Specify when to collect value. Select either and/or all:

   - `When value is ...`

     If you are setting up a request for an asynchronous monitor, this is the only option available.

   - `When value changes`
   - `At each interval`

     Select this option to send an event periodically, regardless of the value.

     Define a polling interval that is appropriate to your system performance and reaction time needs. See Step 3.

3. Specify a polling interval for how often the monitor checks the resource and reports the value.

   This applies only to non-asynchronous monitors and goes with the `At each interval` option in Step 2.

4. Specify how often the monitor should check and send notification about the resource:

   the

   - `Initial` option immediately checks and returns the resource value regardless of threshold conditions
   - the `Repeat` option checks and returns the resource value at each polling interval if threshold conditions have been met
   - the `Return` option checks and returns the resource value after a threshold condition has been resolved and the threshold condition is not longer true.

5. Specify the notification protocols:

   - opcmsg (IT/O), by severity or map severity from values listed
   - TCP or UDP
   - SNMP trap, by severity or map severity from values listed
   - email

- console
- syslog
- textlog

EMS and Target Applications

Target applications receive notification messages about the monitored resources.

To help configure your Network Node Manager and IT/Operations or other system management software for EMS, refer to the *Writing Monitors for the Event Monitoring Service (EMS)* (HP Part Number B7611-90016)*)* developer's kit web page:

1. Go to the web site:

   `http://software.hp.com`.

2. From the web site select `High Availability`, then select *Event Monitoring Service Developers Kit*.

3. Select `Templates`.

Resource Monitors

Resource monitors are applications written to gather and report information about specific resources on the system.

The resource monitor:

- Provides a list of resources that can be monitored
- Provides information about the resources
- Monitors the resources it supports
- Provides values to the EMS API notification

  The EMS framework evaluates the data to determine if an event has occurred. If an event has occurred, the EMS API sends notification in the appropriate format to the configured target(s).

Installing Resource Monitors

To obtain additional information about installed monitors:

1. Go to the `/etc/opt/resmon/dictionary` directory.

   Each monitor registered with EMS has a dictionary file that is stored in this directory.

2. View the monitor dictionary file.

   Each dictionary file name is descriptive of its monitor. The file extension is typically `.dict`. For example, the mibmonitor dictionary filename is `mibmond.dict`.

Configuring Resource Monitors

To configure your HA Monitors use either the Event Monitoring Service (EMS) through SAM or the package configuration area of ServiceGuard GUI or edit the package configuration ascii file.

Writing Resource Monitors

The EMS API provides a method for writing new resource monitors. To create your own monitor, read the *Writing Monitors for the Event*

Monitoring Service (EMS) (HP Part Number B7611-90016) manual and install the developer's kit. Both are available at the following web site:

1. Go to the web site:

   `http://www.software.hp.com`.

2. From the web site select `High Availability`, then select the *Event Monitoring Service Developers Kit*, which includes:

   - *Writing Monitors for the Event Monitoring Service (EMS)* (HP Part Number B7611-90016)
   - ReadMe, Installation, and ReleaseNotes
   - Developers Kit

EMS Framework Components

This section describes the EMS framework components.

The EMS API

The EMS API is the interface between the registrar, client applications, target applications, and resource monitors as illustrated in Figure 1-1. The EMS API is provided as part of the EMS product.

The EMS API manages these events:

- client to registrar communication puts clients in contact with the appropriate monitor for discovery and registering monitor requests.
- registrar to monitor communication passes client requests to appropriate monitor
- make comparisons between the current resource values and pre-selected threshold values
- monitor to target application communication:
  - — sends events to configured targets (pre-existing targets or target you create)
  - — sends notifications to target applications when the resource values meet event criteria

  For exarmple, a target TCP application uses EMS API to translate TCP messages into EMS objects. This enables the fields to be real. The target application then reads the fields of the EMS objects.

The registrar

The registrar is a link between the client applications and the resource monitors. It communicates with the resource monitors on behalf of the client applications to retrieve information requested by the clients. The registrar runs on the same system as the resource monitors. The registrar is provided as part of the EMS product.

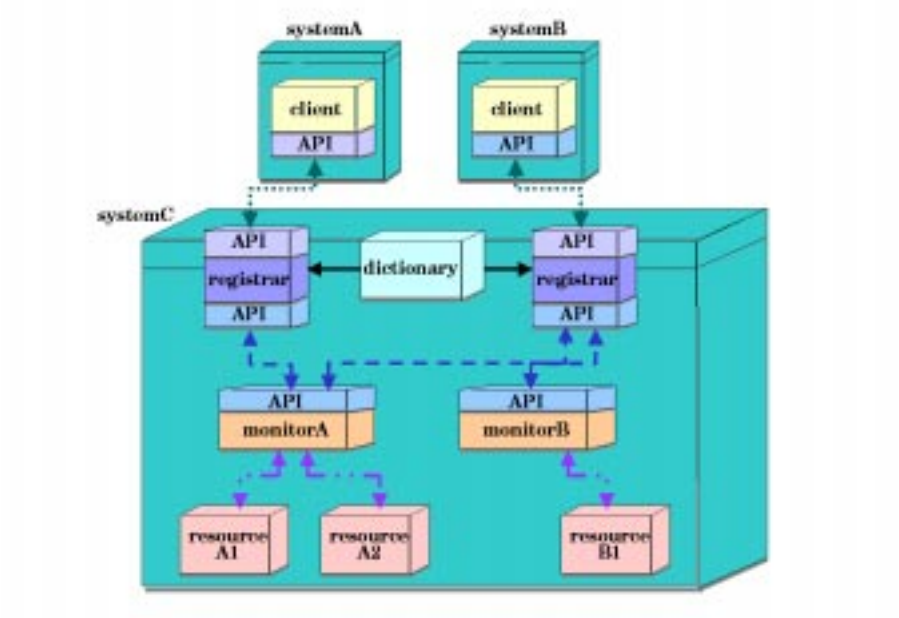
The registrar does not need to keep any state information and does not need to be highly available. It does not need to be running while a resource is being monitored. The registrar is needed only to start the

monitors and to provide communication between clients and monitors.

One registrar process is started each time a client application calls `rm_client_connect()`, so a registrar is always connected to one client. Depending on the requests sent by the client, the registrar may be connected to 0, 1, 2, or more resource monitors concurrently. The information in the messages contains enough information to allow the registrar to route the requests and replies correctly.

Figure 1-3 gives an example of what kinds of connections are possible.

Figure 1-3 Connections Among Clients and registrars



Each time the registrar starts, it reads the resource dictionary, exchanges internal version information with the client application, and prepares to receive client requests. When a request arrives, the registrar analyzes it to determine if it is one that it can reply to, or whether it needs to pass the request to a resource monitor.

When the registrar needs to pass the request to a resource monitor, it needs to determine if the resource monitor is currently running. If the appropriate resource monitor process is not found, the registrar starts the process and waits until the resource monitor can communicate with the registrar.

The Resource Dictionary

The resource dictionary is the mechanism by which the resource monitor identifies itself to EMS. The purpose of the resource dictionary is to give a preliminary picture of the resource structure on a given system. Its main function is to indicate to the registrar which resource monitors should be contacted when information is needed about a certain resource. The resource dictionary defines resources on the local system.

2 Selecting Resources to Monitor

This chapter describes the following:

- Starting the Event Monitoring Service
- Selecting Resources
- Viewing Resource Descriptions

Starting the Event Monitoring Service

To start EMS:

1. Log on as root to the system with EMS and start the graphical version of SAM. From your command line, type:

   sam

2. Double-click the Resource Management icon.
3. Double-click on the Event Monitoring Service icon.

   The main screen with the Actions menu open, shown in Figure 2-1, shows all requests configured on that system.

   If there are no requests and Sentinel monitors are not installed:

   - A message displays on your screen:

     ```
     Currently no resources are being monitored. Use the
     <Add Monitoring Request> action.
     ```

   - The field area of the main screen is empty.

   If Sentinel monitors are installed, the screen is simply blank.

Figure 2-1 Event Monitoring Service Main Screen

Selecting Resources

Resources are divided into classes. To select a resource to monitor:

1. From the `Event Monitoring Service` main screen, click on the Actions menu.

   Refer to the section, “Starting the Event Monitoring Service” on page 30 for instructions on starting EMS.

2. Select Add Monitoring Request

   The top-level resource classes for all installed monitors are dynamically discovered and then listed as shown in Figure 2-2.

   Some Hewlett-Packard products include their own monitors within their product hierarchy. For example, ATM Adapter for HP/9000 Servers, HP OTS 9000 or STM (Support Tools Manager) for HP 9000 hardware monitoring. If this type of product is installed on the system, then its top-level resource class also appears in the `Add or Copy Monitoring Request` screen.

NOTE After installation some monitors must be enabled before their resource classes appear in the EMS `Add or Copy Monitoring Request` screen. For example, the STM hardware monitors. Refer to the documentation for your monitor for instructions on enabling or starting your monitor.

Similarly, top-level resource classes belonging to user-written monitors, created using the procedures described in *Writing Monitors for the Event Monitoring Service (EMS)* (HP Part Number B7611-90016), are discovered and displayed here.

To obtain additional information about any particular monitor:

- Review the monitor dictionary file:

  a. From the command line, go to the `/etc/opt/resmon/dictionary` directory.

     Information about each monitor can be found in the `.dict` files.

  b. View the monitor dictionary file.

The file name corresponds to its monitor. The file extension is `.dict`. For example, the MIB Monitor dictionary filename is `mibmond.dict`.

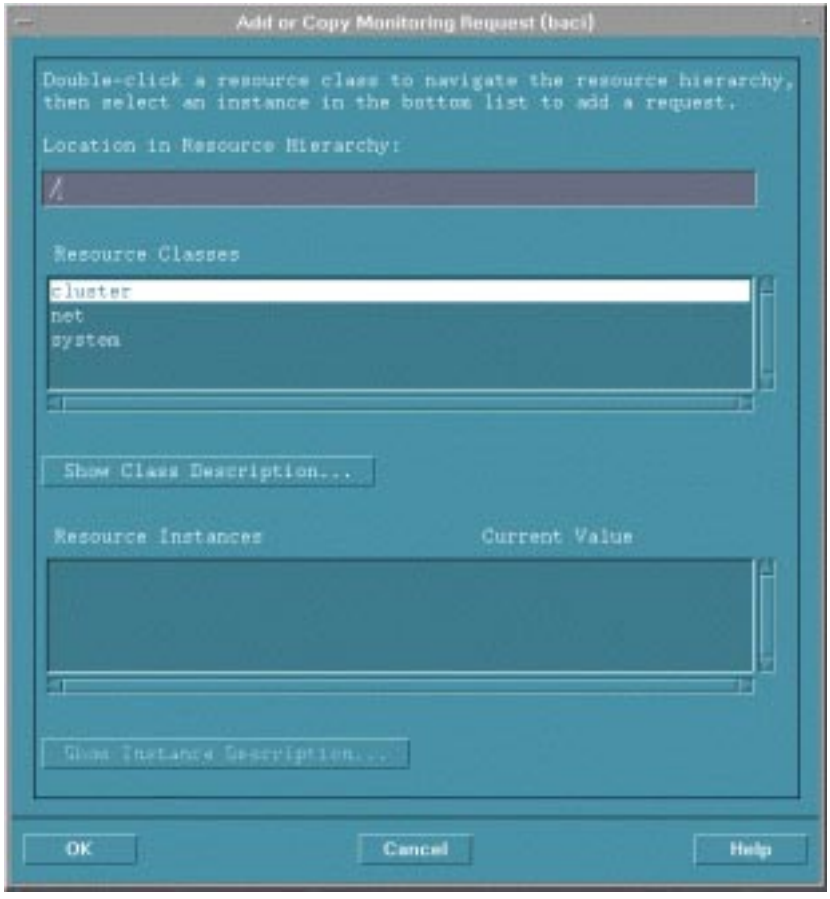
- Review the man page.

  The man page name can be found in the dictionary file with the monitor's name. If a man page was created it is listed in the `MONITOR` entry section of the dictionary file.

- View the resource class or instance description through EMS.

  The `EMS Monitoring Request Parameters` screen has a `View Resource Description` button that displays additional information supplied by the selected resource. See Figure 2-2.

Figure 2-2 EMS Monitoring Request Parameters Screen



3. Double-click on a resource class.

   When you monitor a resource, you actually monitor one or more specific instances of its resource class. View the resource instances associated with the selected resource class in the `Resource Instance` field. See Figure 2-3.

   If the resource class has subclasses, those subclasses are listed in the Resource Classes field.

   Asynchronous monitors are event-driven, rather than polled. They generate messages as events occur, without regard for relative importance. Therefore, if the resource instance is an asynchronous monitor, the Current Value field does not apply, and the field displays

n/a.

Figure 2-3 Add or Copy Monitoring Request Screen

4. Select a specific instance or the wildcard (All Instances).

   The (*) wildcard is a convenient way to create many requests at once. Most systems have more than one disk or network card, and many have several disks. To avoid having to create a monitor request for each disk, select `*(All Instances)` in the Resource Instance box. The `*(All Instances)` listing is always the first item on the list. See the figure above.

   Wildcards are available only when all instances of a subclass are of

the same resource type and there are multiple instances. Selecting the wildcard applies the monitor to all the instances of that resource type. Wildcards are not available for resource classes. For example, a wildcard is available for the status instances in the subclass, `/system/filesystem/availMb`. A wildcard is not available for the entire volume group resource class, `/vg`.

5. Click OK.

   You see the `Monitoring Request Parameters` screen. See Chapter 4, “Defining a Monitoring Request.”

Viewing Resource Descriptions

Resource class and resource instance descriptions are available for each resource.

To see a resource class description, click the Show Class Description button from the Add or Copy Monitoring Request screen.

To see a resource instance description, click either the Show Instance Description button from the Add or Copy Monitoring Request screen or the Show Instance Description button from the Monitoring Request Parameters screen.

3 Defining a Monitoring Request

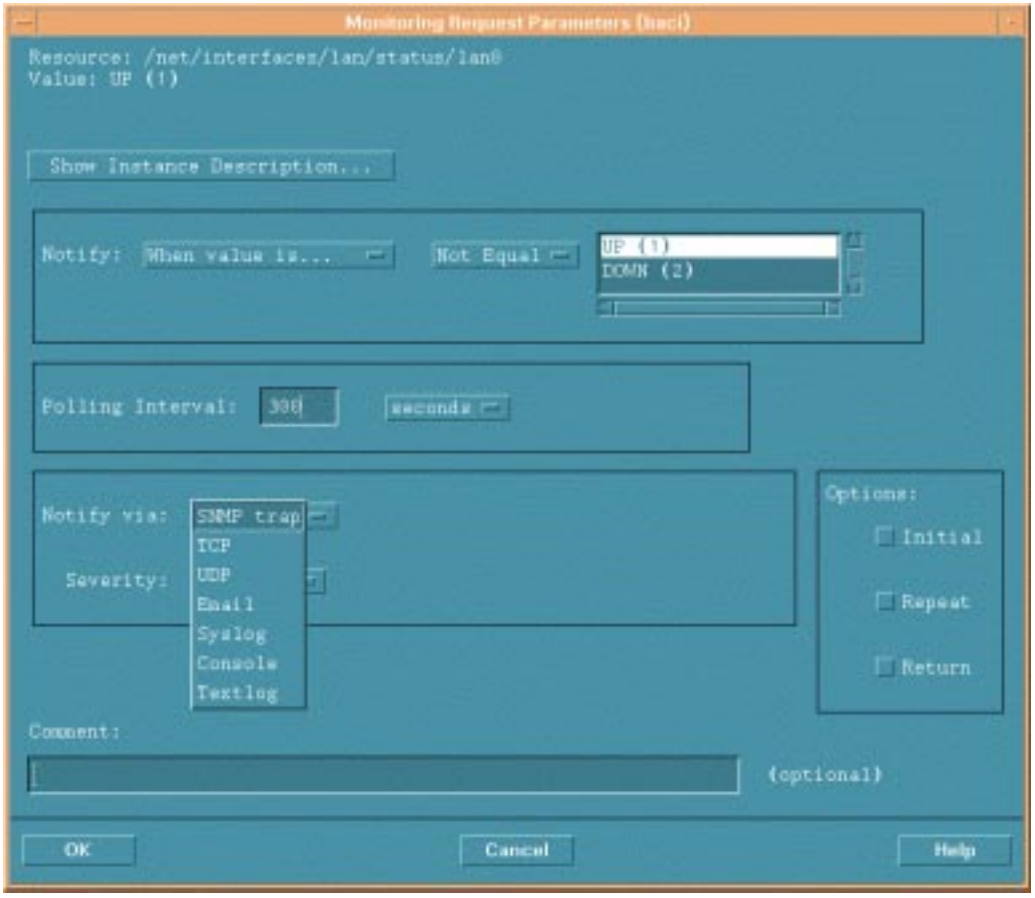
This chapter describes the following:

- Starting a Monitoring Request
- Specifying When to Send Event Notifications
- Setting the Polling Interval
- Setting Event Value Options
- Selecting Protocols for Sending Events
- Adding a Notification Comment

Starting a Monitoring Request

After you have selected a resource to monitor, use the `Monitoring Request Parameters` screen to specify when and how to send event notification (Figure 3-1). The following sections describe the monitoring parameters and provide examples of common applications.

Figure 3-1 Monitoring Request Parameters Screen



Specifying When to Send Event Notifications

When you create a request, you specify the conditions under which you want to collect resource status values. While the monitor may be polling disks every five minutes, for example, you may only want to be alerted when something happens that requires your attention. Specify these conditions in the `Notify` area of the `Monitoring Request Parameters` screen. Here are the terms under which you can be notified:

When value is ...	You define the conditions under which you wish to be notified for a particular resource. Choose an *operator* (=, not equal, >, >=, <, <=) and one of the possible *values* returned by the monitor. The content of the `value` field varies depending upon the resource monitored. The `value` field might display as a list box or a blank field. If a list box displays, use the scroll bar and select from the list. If a `numeric` field displays, type in a value.
When value changes	This sends notification every time a resource value *changes*. Typically this is used for resources whose values change infrequently. Change here means any change. For example, you could receive notification every time the number of mirrored copies of data changes, whether it is from 2 to 1 or from 1 to 2.
At each interval	This sends notification at each polling interval. It is commonly used for reminders or for gathering data for system analysis. Use this for only a small number of resources at a time, and with long polling intervals of several minutes or hours; there is a risk of affecting system performance.

To set an event trigger:

- Select from the listed options in the `Notify` area (`When value is...`, `When value changes`, or `At each interval`).

  Asynchronous monitors are event-driven, rather than polled. They generate messages as events occur. Therefore, if the request is for an asynchronous monitor, only the `When value is...` option is available.

NOTE

Updated monitors may have new status values that change the meaning of your monitoring requests, or generate new alerts. For example, assume you have a request for notification if status > 3 for a resource with a values range of 1 through 7. You receive alerts each time the value equals 4, 5, 6, or 7. If the updated version of the monitor has a new status value of 1 through 8, you also see alerts when the resource equals 8.

Setting the Polling Interval

The polling interval specifies how often the resource monitor checks the resource value. The polling interval is the maximum amount of elapsed time before a monitor knows about a change in status for a particular resource.

The shorter the polling interval, the more likely you are to have recent data. However, depending on the monitor, a short polling interval may use too much CPU and system resources. You need to weigh the advantages and disadvantages between being able to quickly respond to events and maintaining good system performance. Some considerations include:

- The minimum polling interval depends on the monitor's ability to process quickly. For most resource monitors the minimum is 30 seconds. Disk monitor requests can be as short as 10 seconds.
- MC/ServiceGuard monitors resources every few seconds. You may want to use a short polling interval (30 seconds or less) when it is critical that you make a quick failover decision.
- You may want a polling interval of 5 minutes or so for monitoring less critical resources.
- You may want to set a very long polling interval (4 hours) to monitor failed disks that are not essential to the system, but which should be replaced in the next few days.

Asynchronous monitors are event-driven, not polled. They generate messages as events occur. Therefore if the resource is an asynchronous monitor, the `Polling Interval` field displays `n/a`.

To set the Polling Interval:

1. Specify the quantity of time in the `numbered` field.
2. Select the unit of time from the `unit of measure` field list (`seconds`, `minutes`, `hours`, `day`). The maximum value is one (1) day.

Setting Event Value Options

If you select the When value is... from the list in the Notify area, the Options area displays three choices. Select one or more of these three options:

Initial	Use this option to establish a baseline when monitoring resources such as available filesystem space or system load. It can also be used to test whether newly requested events are being sent.
Repeat	Use this option for urgent alerts. The Repeat option sends an alert at each polling interval as long as the notify condition is met. Use this option with caution; there is a risk of high CPU use or filling log files and alert windows.
Return	Use this option to track when emergency situations return to normal.

To set the frequency of the trigger:

- Click one or more buttons on the list in the Options area (Initial, Repeat, and Return).

  Asynchronous monitors are event-driven, not polled. They generate messages as events occur. Therefore if the resource is an asynchronous monitor, the values in the Options area defaults to Repeat and cannot be changed.

  This Options area does not display if you have selected When value changes or At each interval from the list in the Notify area. In these cases the options default to Initial and cannot be changed.

Selecting Protocols for Sending Events

Through the Notify via area specify the protocol you want the monitor to use to send events. The options are described in the following sections.

opcmsg (ITO) Option

This option sends messages to ITO applications via the opcmsg daemon. For this option to display, IT Operation Managed Node Software 3.x or 4.x must be installed on the resource server running HP-UX version 10.20. This option is not currently available on systems running HP-UX version 11.0.

The ITO message severity options are:

- Map from value (this is not available on all monitors)
- Critical
- Major
- Minor
- Warning
- Normal

A specified severity other than Normal is returned under the following conditions:

- When value is . . . If this notification option is set, a non-normal severity occurs when the value changes from FALSE to TRUE.

  For example if a disk is being monitored, you want notification when the disk is down. The When value is condition is FALSE while the disk is up and running correctly. The condition becomes TRUE, meaning action needs to be taken, when the disk is down or not operating correctly.

- When value changes If this notification option is set, a non-normal severity occurs when the current value does not match the previous value

Certain monitors can map directly to OPC severity levels. This is not available with all monitors. Select Map from value option from the list in the Severity area.

If opcmsg is selected, EMS sets the following fields:

- ITO application group: EMS (HP)
- message group: HA
- object: to the full path of the resource being monitored

See *HP OpenView IT/Operations Administrators Task Guide* (Part Number B4249-90003) for more information.

Templates for configuring IT/O and Network Node Manager to display monitored events can be found on the Hewlett-Packard web page at **http://www.software.hp.com**. Click on High Availability, then *Event Monitoring Service Developer's Kit*.

To set the opcmsg protocol for ITO:

1. Specify the notification type from the list in the Notify area.
2. Select the opcmsg (ITO) option from the list in the Notify area.
3. Select the severity from the list in the Severity area:
   - Map from value
   - Critical
   - Major
   - Minor
   - Warning
   - Normal

TCP and UDP Options

This sends TCP or UDP encoded events to the target host name and port indicated for that request. Thus, the message can be directed to a user-written socket program.

To set the TCP or UDP conditions:

1. Select the TCP or UDP option, as appropriate, from the list in the Notify via area.
2. Specify the target host name and the port in their respective fields.

SNMP Traps Option

This sends messages to applications, such as Network Node Manager that use SNMP traps. See *HP OpenView Using Network Node Manager* (P/N J1169-90002) for more information on configuring SNMP traps. Table 3-1 lists traps used by EMS:

Table 3-1 SNMP Traps

Trap Name	Trap Value	Description
EMS_ ENTERPRISE_ OID	"1.3.6.1.4.1.11.2.3.1.7"	
EMS_NORMAL_ OID	"1.3.6.1.4.1.11.2.3.1.7.0.1"	Normal Event
EMS_ABNORMAL_ OID	"1.3.6.1.4.1.11.2.3.1.7.0.2"	Problem Event
EMS_REBOOT_ OID	"1.3.6.1.4.1.11.2.3.1.7.0.3"	Reboot Event
EMS_RESTART_ OID	"1.3.6.1.4.1.11.2.3.1.7.0.4"	Restart Event
EMS_NORMAL_ SEV_OID	"1.3.6.1.4.1.11.2.3.1.7.0.5"	Problem Event w/Normal Severity
EMS_WARNING_ SEV_OID	"1.3.6.1.4.1.11.2.3.1.7.0.6"	Problem Event w/ Warning Severity
EMS_MINOR_ SEV_OID	"1.3.6.1.4.1.11.2.3.1.7.0.7"	Problem Event w/Minor Severity

Table 3-1 SNMP Traps

Trap Name	Trap Value	Description
EMS_MAJOR_SEV_OID	"1.3.6.1.4.1.11.2.3.1.7.0.8"	Problem Event w/Major Severity
EMS_CRITICAL_SEV_OID	"1.3.6.1.4.1.11.2.3.1.7.0.9"	Problem Event w/ Critical Severity

The Severity area options for SNMP traps are:

- Map from value (this is not available with all monitors)
- Critical
- Major
- Minor
- Warning
- Normal

A specified severity other than Normal is returned under the following conditions:

- When value is . . . If this notification option is set, a non-normal severity occurs when the value changes from FALSE to TRUE.

  For example if a disk is being monitored, you want notification when the disk is down. The When value is condition is FALSE while the disk is up and running correctly. The condition becomes TRUE, meaning action needs to be taken, when the disk is down or not operating correctly.

- When value changes If this notification option is set, a non-normal severity occurs when the current value does not match the previous value

Certain SNMP monitoring requests can map directly to severity levels. Select the Map from value option from the list in the Severity area.

To set the SNMP trap:

1. Specify the notification type from the list in the Notify area.
2. Select the SNMP trap option from the list in the Notify via area.
3. Select the severity from the list in the Severity area:
   - Map from value
   - Critical
   - Major
   - Minor
   - Warning
   - Normal

Email Option

This sends event notification to the email address indicated for that request.

To set for an email notification:

1. Select the email option from the list in the Notify via area.
2. Specify the full email address in the email address field.

Console Option

This sends event notification to the system console.

To set for a console notification:

- Select the console option from the list in the Notify via area.

Syslog Option

This sends event notification to the system log.

An abnormal event message (error) is returned under the following conditions:

- When value is . . . If this notification option is set, a non-normal severity occurs when the value changes from FALSE to TRUE.

  For example if a disk is being monitored, you want notification when the disk is down. The When value is condition is FALSE while the

disk is up and running correctly. The condition becomes TRUE, meaning action needs to be taken, when the disk is down or not operating correctly.

- When value changes If this notification option is set, a non-normal severity occurs when the current value does not match the previous value

For an abnormal event, a system logging level of error will be associated with the logged message.

To set for a system log notification:

- Select the syslog option from the list in the Notify via area.

Textlog Option

This sends event notification to a file you specify.

To set for a textlog notification:

1. Select the Textlog option from the list in the Notify via area.
2. Specify the filename and path in the File Path field.

   The default path is /var/opt/resmon/log/event.log.

Adding a Notification Comment

The notification comment is useful for sending task reminders to the recipients of an event. For example, you can configure a disk monitor request that reports an alert when an entire mirror has failed. When that event shows up in IT/Operations, you may want a notification comment to include the name of the person to contact.

4 Changing Monitoring Requests

This chapter describes the following:

- Copying Monitoring Requests
- Modifying Monitoring Requests
- Removing Monitoring Requests
- Viewing Monitoring Requests

Copying Monitoring Requests

There are two ways to use the copy function:

- To create requests for *multiple* resources using the *same* monitoring parameters.
- To create requests for the *same* resource using *different* monitoring parameters.

To create requests for multiple resources using the same monitoring parameters:

1. From the Event Monitoring Service main screen, select the monitoring request whose parameters you wish to copy.

   You need to have configured at least one similar request for a similar instance.

2. Select Actions menu: Copy option.

   You see the Add or Copy Monitoring Request screen.

3. From the Add or Copy Monitoring Request screen, select a different resource instance and click OK.

   You see the Monitoring Request Parameters screen.

4. Click OK in the Monitoring Request Parameters screen.

   You see a message that indicates the new request has been added. You see the Event Monitoring Service main screen.

To create requests for the same resource using different monitoring parameters:

1. From the Event Monitoring Service main screen, select the monitoring request with the instance for which you wish to have multiple monitoring requests.

   You need to have configured at least one request for the instance.

2. Select Actions menu: Copy option.

   You see the Add or Copy Monitoring Request screen.

3. Click OK in the Add or Copy Monitoring Request screen.

   You see the Monitoring Request Parameters screen.

4. Modify the parameters as desired in the `Monitoring Request Parameters` screen.
5. Click `OK`.

   You see a message that indicates the new request has been added. You see the `Event Monitoring Service` main screen.

Modifying Monitoring Requests

To change the monitoring parameters of a request:

1. From the `Event Monitoring Service` main screen, select the monitoring request you want to modify and either:
   - Double-Click the request, or
   - Select `Actions` menu: `Modify`

   You see the `Monitoring Request Parameters` screen.
2. Modify the parameters as desired, by editing the fields in the `Monitoring Request Parameters` screen.
3. Click `OK`.

   You see a message that indicates the request has been modified. You see the `Event Monitoring Service` main screen.

Removing Monitoring Requests

You can remove one or more requests using the Remove Monitoring Requests option. To remove monitoring requests:

1. From the Event Monitoring Service main screen, select the monitoring request you wish to remove.

   To select contiguous multiple requests, hold the Shift key and click.

   To select individual multiple requests, hold the Ctrl key and click.

2. Select Actions menu: Remove option.

   You see a Confirmation screen.

3. Click OK on the Confirmation screen.

   You see a message that indicates how many request(s) have been removed. You see the Event Monitoring Service main screen.

4. To start monitoring the resource again you must recreate the request, either by copying a similar request for a similar resource or by re-entering the data.

Viewing Monitoring Requests

To view the parameters for a monitoring request:

1. From the Event Monitoring Service main screen, select the monitoring request you wish to view.
2. Select Actions menu: View

   You see the View Monitoring Request Parameters screen with the parameters specified for the monitoring request.
3. To modify the parameters of this request, click the Modify Monitoring Request option. You see the Monitoring Request Parameters screen. Proceed as described in "Modifying Monitoring Requests."
4. To close the View Monitoring Request screen, click OK.

5 Monitoring ServiceGuard Package Dependencies

This chapter describes how to use SAM to define package dependencies on EMS resources. ServiceGuard by itself automatically monitors specific resources. Using ServiceGuard with EMS adds to the list of resources that can be monitored. These resources need to be configured and identified to ServiceGuard as package resource dependencies.

You create a monitoring request to observe the EMS resource and to notify ServiceGuard when that resource reaches a critical user-defined level. At that time ServiceGuard will fail over the package. The following are some examples of how EMS might be used:

- In a cluster where one copy of data is shared between two nodes (both configured with EMS), you may want to fail a package over when, for example, the LAN or SCSI host adapter fails on the node running the package. ServiceGuard compares the resource `UP` values on other configured nodes, and fails the package over to the node that has the correct resources available.
- In a cluster where each node has its own copy of data, you may want to failover a package to another node for any number of reasons:
  - host adapter, bus, controller, or disk failure
  - unprotected data (the number of copies is reduced to one)
  - degraded performance because one of the PV links has failed

This information for creating requests is also valid for EMS sold with other products (ATM, OTS, HyperFabric, or STM hardware monitors, for example) and for user-written monitors written according to developer specifications in *Writing Monitors for the Event Monitoring Service (EMS)* (HP Part Number B7611-90016).

NOTE Create the same requests on all nodes configured for an ServiceGuard package.

A package can depend on any resource whose monitor is registered with EMS. To create package dependencies:

1. Halt the cluster. Include a force option to stop all packages, by typing:

   cmhaltcl -f

   To add an EMS resource, the ServiceGuard cluster must be down. You can modify existing EMS resources through ServiceGuard while the cluster is running.

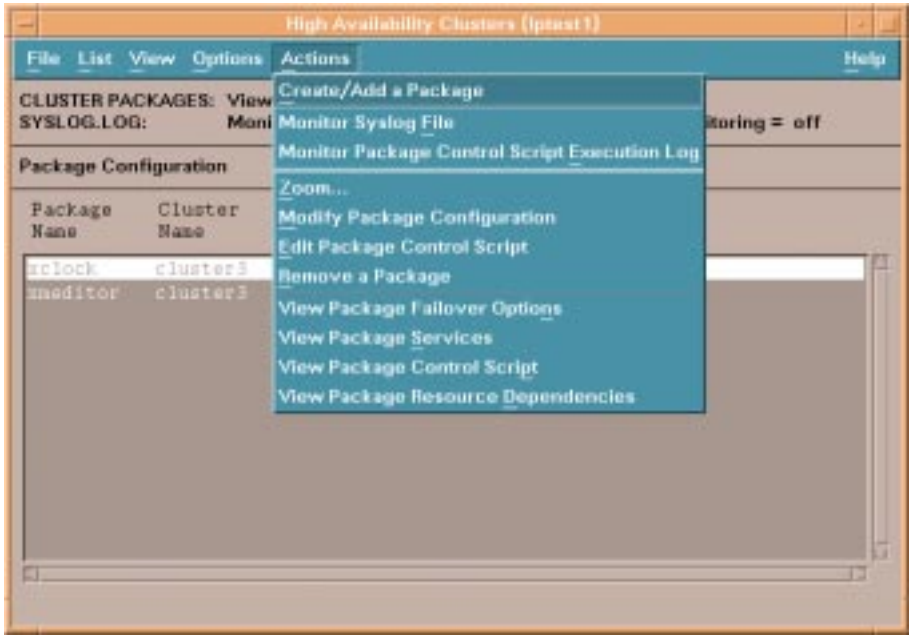
2. From your command line, start SAM, by typing:

   sam

3. Double-click the Clusters icon.
4. Double-click the High Availability Clusters icon.
5. Double-click on the Package Configuration icon.

   The High Availability Clusters screen shows all requests currently configured on that system. If there are no requests currently configured, the field area is empty. Figure 5-1 shows the High Availability Clusters screen with the Actions menu displayed.

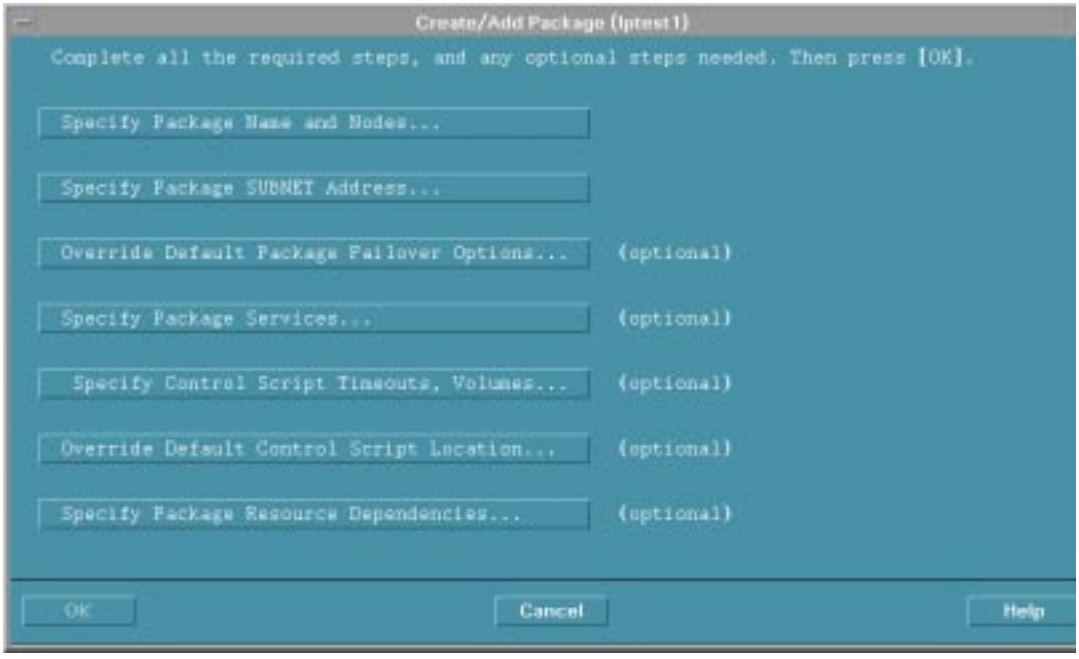
Figure 5-1 High Availability Clusters Screen



6. From the Actions menu, select either the `Create/Add a Package` or `Modify Package Configuration` option.

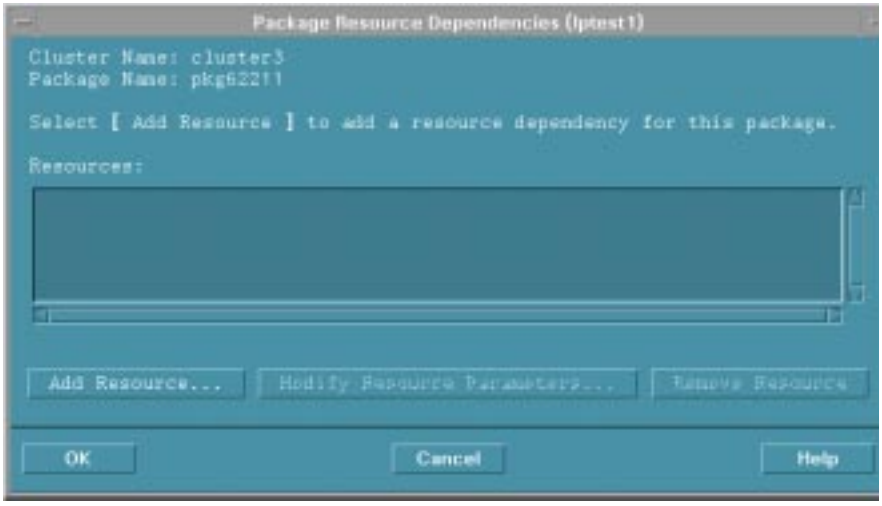
   If you select `Create/Add a Package`, a screen similar to Figure 5-2, displays.

Figure 5-2 Create/Add a Package Screen



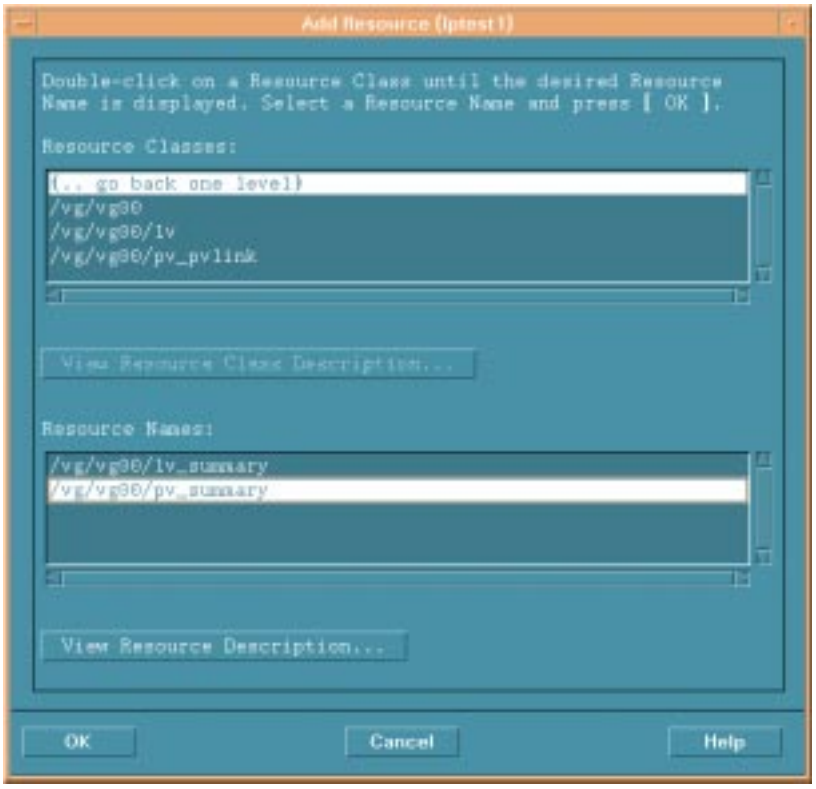
7. If you have not previously done so, click `Specify Package Name and Node` and `Specify Package SUBNET Address`. Then click on `Specify Package Resource Dependencies...` to add EMS resources as package dependencies. A screen similar to Figure 5-3 displays.

Figure 5-3 Package Resource Dependencies Screen



8. The Resources: field lists all the installed resources discovered by ServiceGuard. To make a package dependent on an EMS resource, select it from the list, then click Add Resource.... An Add Resources screen, similar to Figure 5-4 displays.

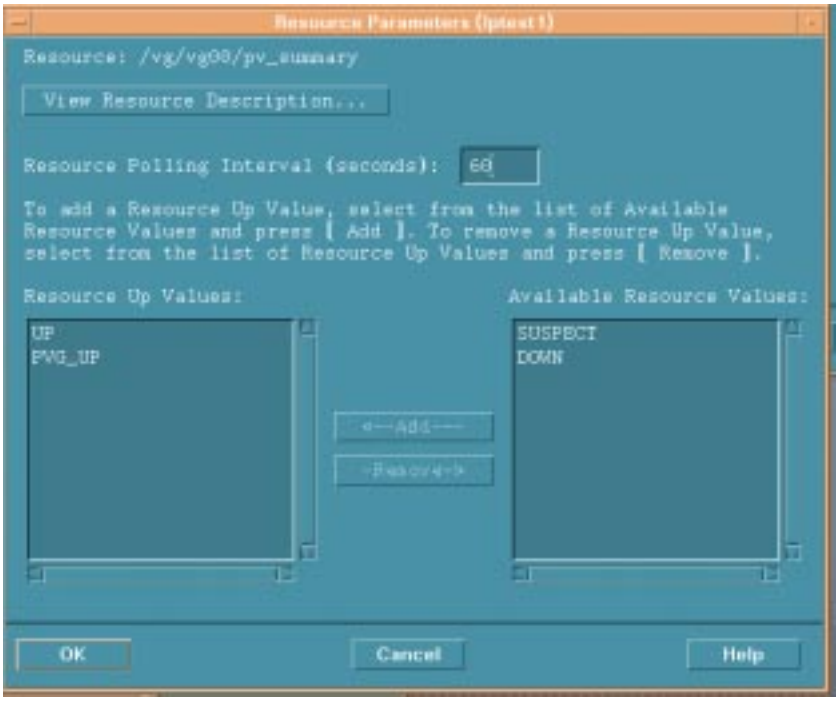
Figure 5-4 Add Resources Screen



9. Click through the Resource Classes and Resource Names to select the entity you wish to monitor. Click OK. A Resource Parameters screen, similar to Figure 5-5 displays.

   The example in Figure 5-5 shows the possible values for pv_summary. Different resources show different available UP values.

Figure 5-5 Resource Parameters Screen



NOTE Make sure you always select UP in the Resources Up Values field. ServiceGuard creates an EMS request that sends an event if the Resources Up Value field is not equal to the UP value.

If you select only UP, the package fails over if the value is anything but UP. In this example, if you select UP and PVG-UP, the package fails over if the pv_summary value is not equal to UP or PVG_UP; in other words, if pv_summary is SUSPECT or DOWN.

The polling interval determines the maximum amount of elapsed time before the monitor knows about a change in resource status. For critical resources, you may want to set a short polling interval, such as 30 seconds, but this could adversely affect system performance. With longer polling intervals you gain system performance, but you risk not detecting problems soon enough.

You can also add resources as package dependencies by modifying the package configuration file. The default filename is `/etc/cmcluster/pkg_name.ascii`. See *Managing MC/ServiceGuard* for details on how to modify this file. For example, the syntax might be:

```
RESOURCE_NAME                /vg/vg01/pv_summary
RESOURCE_POLLING_INTERVAL    60
RESOURCE_UP_VALUE            = UP
RESOURCE_UP_VALUE            = PVG_UP
```

ServiceGuard automatically distributes this information to every node in the cluster when the package configuration file is applied to the cluster configuration (see the `cmapplyconf` command).

6 Monitoring Cluster Resources

The HA Cluster Monitor sends events regarding the status of a cluster. If you have OpenView, we recommend using HP ClusterView to monitor cluster status and receive cluster events. HA Cluster Monitor is primarily for use with non-OpenView systems, for example, CA UniCenter.

The sections in this chapter are:

- Cluster Monitor Reference
- Creating Cluster Monitoring Requests

Cluster Monitor Reference

The HA Cluster Monitor is useful in environments not running HP OpenView ClusterView. The HA Cluster Monitor reports information on the status of the cluster to which the local node belongs.

The resources monitored are:

- `/cluster/status/`*`clustername`*, a summary of the state of all nodes in the cluster *`clustername`*
- `/cluster/localNode/status/`*`clustername`*, the status of a given node in a cluster. A node can only be a member of one cluster
- `/cluster/package/package_status/`*`packageName`*, the status of a ServiceGuard package in a cluster
- `/cluster/package/service_status/`*`packageName/serviceName`*, the status of a ServiceGuard service in a cluster package

The HA Cluster Monitor is part of the MIB Monitors package. Table 6-1 lists the HA Cluster Monitors.

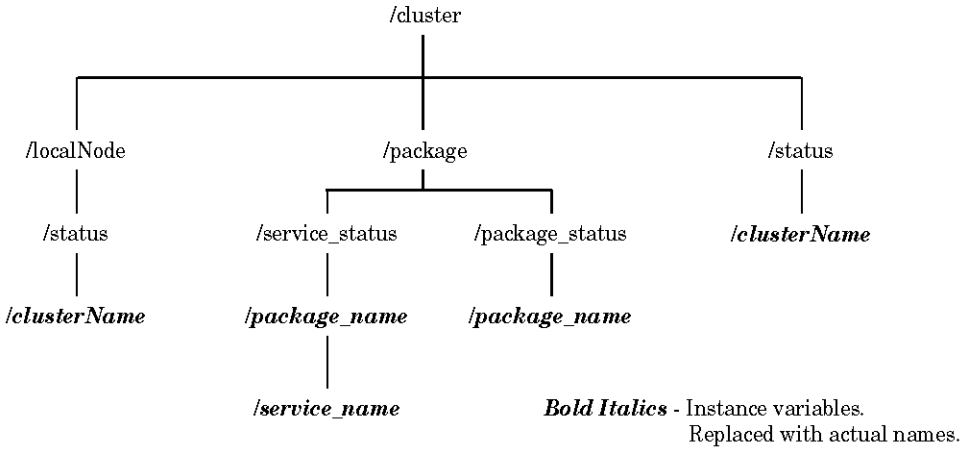
Table 6-1 HA Cluster Monitor Names and Resources

Monitor	Resource Type	Resource
clustermond	Cluster	`/cluster/status/<clusterName>`
	Local Node	`/cluster/localNode/status/ <clusterName>`
pkgmond	Package	`/cluster/package/ package_status/<packageName>`
svcmond	Service	`/cluster/package/ service_status/<packageName>/<s erviceName>`

To fix any problems detected by the cluster monitor, refer to *Managing MC/ServiceGuard* (HP Part Number B3936-90026).

Figure 6-1 shows the cluster monitor class hierarchy.

Figure 6-1 Cluster Monitor Resource Class Hierarchy



Items in boxes are resource instances that can be monitored. Variables in italics change depending on the names of the clusters and packages on the system.

Cluster Status

The cluster status is the status of the MC/ServiceGuard cluster to which this node belongs.The status is from the perspective of the node for which the request was created.

The hp-mcCluster MIB variable, `hpmcClusterState`, provides the cluster status information to the monitor.

The `cmviewcl -v` command displays detailed information about the current status of the cluster and packages on the cluster.

Table 6-2 Interpreting Cluster Status

Resource Name: /cluster/status/*clusterName*		
Condition	**Value**	**Interpretation**
UP	1	The node can access the cluster.
UNKNOWN	2	The node may be separated from other active cluster elements (for example the heartbeat LAN) and has insufficient information to tell if the cluster is accessible.

Table 6-2 Interpreting Cluster Status

Resource Name: /cluster/status/*clusterName*		
Condition	**Value**	**Interpretation**
DOWN	3	The node cannot access the cluster.

You might request to be notified when the cluster is not up. You could then verify whether the cluster was shut down intentionally.

The minimum polling interval for cluster status is 30 seconds. You may want a longer interval, especially if system performance is affected.

Node Status

The node status is the current status of a node relative to a particular cluster.

The hp-mcCluster MIB variable, `hpmcNodeStatus`, provides the node status information to the monitor.

The `cmviewcl -v` command displays detailed information about the current status of the cluster and packages on the cluster.

Table 6-3 Interpreting Node Status

Resource Name: /cluster/localNode/status/*clusterName*		
Condition	**Value**	**Interpretation**
`RUNNING`	1	Node is accessible and operating normally.
`INITIALIZING`	2	Node's cluster daemon has started, but is not ready to communicate with other nodes' daemons.
`RECONFIGURING`	3	Node is running protocols to make sure all other nodes agree to the new membership in the cluster.
`INVALID`	4	The cluster status may be DOWN.
`HALTED`	5	Node has been removed from the cluster, with the `cmhaltnode` command, for example.
`FAILED`	6	Node is no longer a member of an active cluster.

You might want to create a request that notifies you when the local node is not running. You can then verify whether the node or MC/ServiceGuard was stopped intentionally.

The minimum polling interval for node status is 30 seconds. You may want a longer interval, especially if system performance is affected.

Package Status

The package status is the status of each package running on this node.

The hp-MCCluster MIB variable, `hpmcSGPkgStatus`, provides the package status information to the monitor.

The `cmviewcl -v` command displays detailed information about the current status of the cluster and packages on the cluster.

Table 6-4 Interpreting Package Status

Resource Name: /cluster/package/package_status/*packageName*		
Condition	**Value**	**Interpretation**
UP	1	The package is running on the local node.
UNKNOWN	2	The package is not running on the local node, but may be running on another node in the cluster.
DOWN	3	The package is not running on any node in the cluster.

You might want to be notified when the value of any of the packages changes to UNKNOWN or DOWN, so you can verify that MC/ServiceGuard successfully migrated the package to another system.

You may see many packages with UNKNOWN status. This is because only the node running a package has complete status for a package. Other nodes often have inactive volume groups that make it impossible to have complete knowledge of package status. If a package is running on another node in the cluster, the current node may not have complete status on that package, and reports the condition UNKNOWN.

The minimum polling interval for package status is 30 seconds. You may want a longer interval, especially if system performance is affected.

Service Status

A service is part of a package. The service status is the status of each service running on this node.

The hp-MCCluster MIB variable, `hpmcSGPkgSvcStatus`, provides the service status information to the monitor.

The `cmviewcl -v` command displays detailed information about the current status of the cluster and services on the cluster.

Table 6-5 Interpreting Service Status

Resource Name: /cluster/package/service_status/*packageName/serviceName*		
Condition	**Value**	**Interpretation**
UP	1	The service is running on the local node.
UNKNOWN	2	The service is not running on the local node, but may be running on another node in the cluster.
DOWN	3	The service is not running on any node in the cluster.

You might want to be notified when the value of any of the services changes to UNKNOWN or DOWN, so you can verify that MC/ServiceGuard successfully migrated the package and its services to another system.

You may see many services with UNKNOWN status. This is because only the node running a package with a service has complete status for a service. Other nodes often have inactive volume groups that make it impossible to have complete knowledge of service status. If a service is running on another node in the cluster, the current node may not have complete status on that service, and reports the condition UNKNOWN.

The minimum polling interval for service status is 30 seconds. You may want a longer interval, especially if system performance is affected.

Creating Cluster Monitoring Requests

For most ServiceGuard or cluster configurations, we suggest creating the following requests on each node in a cluster:

Table 6-6 Recommended Cluster Requests

Resources to monitor	Monitoring Parameters			
	Notify		Value	Option
`/cluster/status/`*`clusterName`*	when value is	not equal	`UP`	`INITIAL`
`/cluster/localNode/status/`*`clusterName`*	when value is	not equal	`RUNNING`	`INITIAL`
`/cluster/package/status/`*`packageName`*	when value is	not equal	`UP`	`INITIAL`
`/cluster/package/service_status/`*`packageName`*`/`*`serviceName`*	when value is	not equal	`UP`	`INITIAL`

The `INITIAL` option is recommended for comparison, so you know the state of all nodes, clusters, and packages when you first start monitoring them.

7 Monitoring Network Interfaces

The HA Network Interface Monitor detects whether your LAN interface is up or down. It allows you to send events to a system management interface as an alternative to looking in `syslog` for LAN status. The HA Network Interface Monitor, `lanmond`, is part of the MIB Monitors package.

The sections in this chapter are:

- Network Monitor Reference
- Configuring Network Monitoring Requests

Network Monitor Reference

The HA Network Interface Monitor provides status on the LAN interfaces in a given node. It monitors all the interfaces that you see when you run the `lanscan` command on a system.

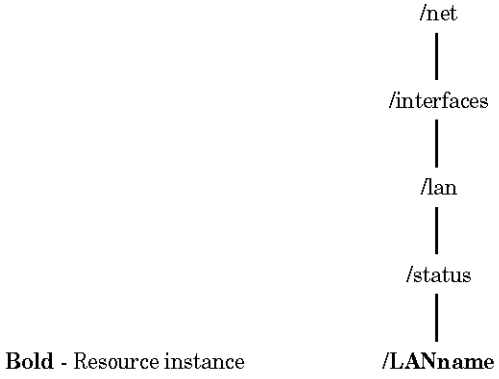
The HA Network Interface Monitor is part of the MIB Monitors package. Table 7-1 lists the HA Network Interface Monitor.

Table 7-1 HA Network Interface Monitor Names and Resources

Monitor	Resource Type	Resource
lanmond	LAN Interface	`/net/interfaces/lan/status/<LANname>`

Figure 7-1 illustrates the network monitored resources.

Figure 7-1 Network Monitor Resource Class Hierarchy



The `MIB-2` variable, `ifOperStatus`, provides the `LAN` interface status to the monitor. The monitor reports `DOWN` if it read the MIB value of TESTING. See Table 7-2.

To verify the operational status of the `LAN` interface, use the `lanscan`

(1M) or lanadmin(1M) commands.

Table 7-2 Interpreting LAN Interface Status

Resource Name: /net/interfaces/lan/status/*LANname*		
Condition	**Value**	**Interpretation**
UP	1	The LAN interface is sending and receiving packets.
DOWN	2	The LAN interface is not passing operational packets.

The EMS product depends on TCP/IP (or UDP) to send events to targets such as HP OpenView IT/Operations or MC/ServiceGuard. If all LAN interfaces on a subnet fail, notifications may not be received by a remote target.

Standby LANs are reported as DOWN unless they have been activated to replace a failed LAN interface.

The minimum polling interval is 30 seconds.

Configuring Network Monitoring Requests

Table 7-3 recommends two monitoring requests for each node. With these requests, you would see events when a LAN card failed, and again when it came back up, and you would see an event every hour that the LAN card was down. You may elect to change the polling interval or not to configure a reminder at all.

Table 7-3 Recommended LAN Interface Requests

Resources to monitor	Monitoring Parameters				
	Notify		Value	Option	Polling Interval
/net/interfaces/lan/status/*LANname*	when value is	<	UP	RETURN	30 sec.
/net/interfaces/lan/status/*LANname*	when value is	=	DOWN	REPEAT	1 hour

8 Monitoring System Resources

The HA System Resource Monitor sends events about the number of users, available file system space, and job queues to help you load-balance and tune your system to keep it available. It is an alternative to reading `syslog` files to get this information. The HA System Resource Monitor, `pkgmond`, is part of the MIB Monitors package.

The sections in this chapter are:

- System Monitor Reference
- Creating System Resource Monitoring Requests

System Monitor Reference

The system monitor reports information on system resources:

- `/system/numUsers`, tells you the number of users on a given node.
- `/system/jobQueue1Min`, `/system/jobQueue5Min`, and `/system/jobQueue15Min`, tell you the number of processes waiting for CPU and performing disk I/O as an average over 1, 5, and 15 minutes respectively. This is the same as the load averages reported by `uptime` (1).
- `/system/filesystem/availMb/`*`fsName`*, tells you the number of megabytes available for use in filesystem *`fsName`*.

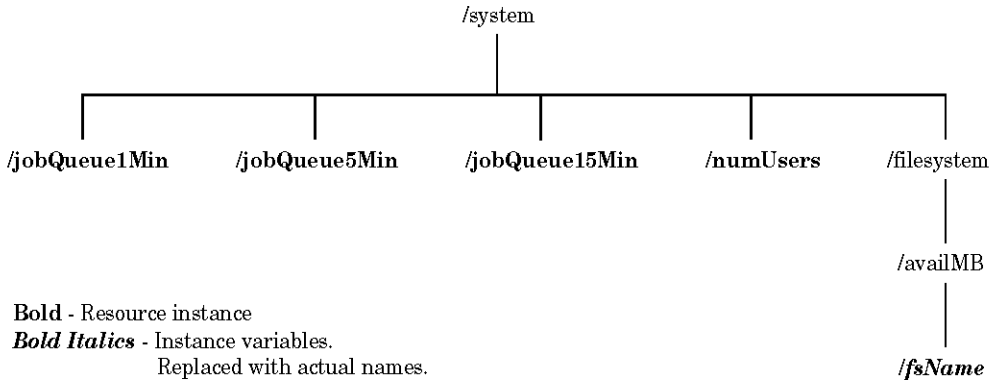
The HA System Resource Monitor is part of the MIB Monitors package. Table 8-1 lists the HA System Resource Monitors.

Table 8-1 HA System Resource Monitor Names and Resources

Monitor	Resource Type	Resource
mibmond	System	`/system/numUsers`, `/system/jobQueue1Min`, `/system/jobQueue5Min`, `/system/jobQueue15Min`
fsmond	Filesystem Available Space	`/system/filesystem/availM`

Figure 8-1 shows the system resource monitor hierarchy. Items in boxes are resource instances that can be monitored. The *`fsName`* in italics changes depending on the names of the file systems.

Figure 8-1 System Resource Monitor Class Hierarchy



Number of Users

The number of users tells you how many users are logged in to a given system.

The MIB variables `computerSystem`, `fileSystemBavail`, and `fileSystemBsize` from the hp-unix MIB provides the resource value to the monitor.

To verify the number of users on the system, use the `uptime` (1) command.

Table 8-2 Interpreting Number of Users

Resource Name	Value Range	Interpretation
/system/numUsers	integer	total number of users logged in to the node

Alerts for number of users can be used to determine the best time to run backups or other maintenance, or for load-balancing, to disallow more than a certain number of users on a given system.

The minimum polling interval is 30 seconds. We recommend a longer interval. Short polling intervals may adversely affect system performance.

Job Queues

The job queue monitor checks the average number of processes that have been waiting for CPU and performing disk I/O over the last 1, 5, or 15 minutes. A value of 4 in `/system/jobQueue5Min` means that at the time of polling there was an average of 4 jobs in the queue over the last 5 minutes.

The MIB variables `computerSystemAvgJobs1`, `computerSystemAvgJobs5`, and `computerSystemAvgJobs15` from the hp-unix MIB provides the resource value to the monitor.

To verify the load averages on the system, use the `uptime` (1) command.

Table 8-3 Interpreting Job Queues

Resource Name	Value Range	Interpretation
/system/jobQueue1Min	integer	average number of jobs in the queue in the last minute
/system/jobQueue5Min	integer	average number of jobs in the queue in the last 5 minutes
/system/jobQueue15Min	integer	average number of jobs in the queue in the last 15 minutes

The minimum polling interval is 30 seconds. Unless your system load tends to fluctuate wildly and need load-balancing attention frequently, set a polling interval greater than or equal to the job queue interval: 1, 5, and 15 minutes, respectively.

Filesystem Available Space

The filesystem monitor checks the number of megabytes available for use in each file system on the node. File systems must be mounted and active to be monitored. File systems mounted over the network, such as NFS file systems, are not monitored.

The MIB variables `fileSystemBavail`, and `fileSystemBsize` from the hp-unix MIB are used to calculate the number of available Kb in the file systems. The number is then divided by 1024 to get the number of available Mb.

Table 8-4 Filesystem Available Space

Resource Name

Most common names are:
 	
`/system/filesystem/availMb/stand`
`/system/filesystem/availMb/root`
`/system/filesystem/availMb/home`
`/system/filesystem/availMb/opt`
`/system/filesystem/availMb/tmp_users`
`/system/filesystem/availMb/usr`
`/system/filesystem/availMb/var`

You may have more file systems, or different names, depending on how you configured file systems on your system. You see that when a file system starts filling up, you can clean up old files or add disk space and reconfigure your file systems.

NOTE Because the "/" character is not valid in a resource name, it is replaced by the "_" character. So the file system `/tmp/users` would appear as the resource name /system/filesystem/availMb/tmp_users. For the same reason the root file system (/) is replaced by the name "root".

The minimum polling interval is 30 seconds. We recommend a longer interval; short polling intervals may adversely affect system performance. When configuring requests through EMS, a wildcard (*) is available to monitor all file systems on a system.

Creating System Resource Monitoring Requests

Table 8-5 shows examples of how you might monitor system resources.

Table 8-5 Examples of System Resource Requests

To be alerted when...	Resources to monitor	Monitoring Parameters			
		Notify		Value	Option
fewer than 5 users are on the node, for running backups	/system/numUsers	when value is	<	5	
more than 20 users are on the node, for load balancing, and when value returns to below 20 users	/system/numUsers	when value is	>	20	RETURN
system load is high	/system/ jobQueue1Min	when value is	>	7	INITIAL
	/system/ jobQueue5Min	when value is	>	4	INITIAL
	/system/ jobQueue15Min	when value is	>	3	INITIAL
file systems are running out of space	/system/filesystem/ availMb for: /home /opt /root /stand /tmp /usr /var	when value is	<	50	INITIAL

The job queue and file system resources have the INITIAL option set to give a baseline for comparison. The job queue threshold value decreases

for the longer job queues because the longer something is in the queue, the more likely it is that a node needs to be load-balanced.

A Dictionary File Command Line Options

This appendix lists the command line options available for the MIB Monitors. Typically, these options may be added to the dictionary file entry that describes how to launch a particular monitor.

MIB Monitor Command-Line Options

Perform additional MIB monitor configuration by using the following command-line options. This includes the HA Cluster Monitor, HA Network Interface Monitor, and HA System Resource Monitor.

Specify one or more of the below listed options in the `MONITOR` statement of the dictionary file for each monitor. For example, within the file: `/etc/opt/resmon/dictionary/mibmond.dict`.

-c *`community`* The community string to use with the SNMP requests. The string `public` is used by default.

-d Turn on internal daemon debugging messages to get a trace. This allows additional information to be logged to the `/var/opt/resmon/log/mibmond.log` file.

CAUTION The `/var/opt/resmon/log/mibmond.log` file grows without bound. Only use the `-d` option if directed to do so by your HP Support Personnel.

-h *`hostname`* Perform SNMP queries against specified hostname. By default the local system is queried to get the contents of the MIB.

-l Write a syslog message every time SNMP is queried for a monitored resource value

-p *`portnumber`* Perform SNMP queries against specified port number. This option is useful when querying proxy SNMP agents. By default SNMP queries are directed against the SNMP port (161)

-r *`retries`* Attempt to retransmit SNMP requests *`retries`* number of times before considering the host unreachable. The default is to retransmit 1 time.

-t *`timeout`* Wait *`timeout`* number of seconds before retransmitting SNMP requests. By default, 5 seconds will expire between retransmissions.

B Troubleshooting

This section gives hints on testing your monitoring requests, and gives you some information about log files and monitor behavior that will help you determine the cause of problems. For information on fixing problems detected by monitors, see the list of related publications in the Preface. This chapter has the following sections:

- EMS Directories and Files
- Logging and Tracing
- Performance Considerations
- Testing Monitor Requests
- MIB Monitor Troubleshooting

EMS Directories and Files

EMS files are located in `/etc/opt/resmon` and `/opt/resmon`. Table B-1 lists files and directories that might help you determine the cause of some problems:

Table B-1 EMS Directories and Files

`/etc/opt/resmon/config`

This file determines how often EMS checks that monitors are running (have not died).

`/etc/opt/resmon/dictionary`

This directory contains resource dictionaries for the various monitors. The cluster, network, and system resource monitors are in the `mibmond.dict`. If you were writing your own monitor, the dictionary would go in this directory.

`/etc/opt/resmon/lbin`

This is the directory where EMS product-specific monitors reside. Some important daemons in the directory:

`p_client` restarts persistent requests for any failed monitors.

`registrar` passes monitoring requests to the correct monitors.

Table B-1 EMS Directories and Files

`/etc/opt/resmon/log`

This is a directory of log files used by EMS:

- `client.log` stores calls made by clients, such as MC/ServiceGuard or the SAM interface to EMS.
- `api.log` stores api calls made by monitors.
- `registrar.log` contains errors found when reading the resource dictionary.
- `emsagent.log` is the SNMP subagent responsible for sending EMS events through an SNMP trap.

`/opt/resmon/bin/resls`

This command can be used to discover and get details on resources available for monitoring.

`/opt/resmon/bin/resdata`

This is used to get additional information from EMS regarding events or monitor restarts.

Logging and Tracing

Use logging for most troubleshooting activities. By default the monitors log to `api.log`. Logging to `/var/adm/syslog/syslog.log` is `ON` by default for the disk monitor and `OFF` by default for the remaining monitors. Tracing should only be used when instructed to do so by HP support personnel. This is not available with all monitors.

EMS Logging

Log files in `/etc/opt/resmon/log/` contain information logged by the monitors.

Look at the `client.log` if you seem to be having a problem with the SAM, or any other client, interfaces to EMS or MC/ServiceGuard. With the default level of logging, only audit and error messages are logged. An example of an audit message is:

```
User event occurred at Thu Jul 31 16:13:31 1997
Process ID: 10404 (client)  Log Level:Audit
+ /vg/vg00/lv/copies/* (8 instances)
If (<1), OpC (m/n), 18000s, Thu Jul 31 16:13:31 1997
```

The plus (`+`) means that request has been added. A minus (`-`) indicates a removal. A minus (`-`) followed by a plus (`+`) indicates a modification. Events sent to targets are marked with period (`.`). Errors are marked with `Log Level: Error` or with `Log Level: Warning`.

Look at the `api.log` if you seem to be having a problem with a specific monitor. Check for warnings or errors.

Some monitors have their own logs, refer to the man page for individual monitors.

Log File Size

EMS log files are normally under `/etc/opt/resmon/log`. Although their size is limited, the EMS log files can grow to 13 MBytes. In addition, some monitors put their own log files in the EMS log directory. EMS does not control the size of monitor log files. If disk space is limited in the file system that contains `/etc/opt/resmon/log`, relocate the EMS log directory and make `/etc/opt/resmon/log` a symbolic link to the new location. To relocate the directory:

```
mkdir /newpath/resmon
mv /etc/opt/resmon/log /newpath/resmon  # create /newpath/resmon/log
                                        # remove /etc/opt/resmon/log
ln -s /newpath/resmon/log /etc/opt/resmon/log
```

NOTE EMS requires that /etc/opt/resmon, the parent directory, reside on the root file system. Do not move all of /etc/opt/resmon to another file system.

High Availability Monitors

High availability monitors provide additional logging support.

NOTE Logging will occur at every polling interval. This can create a very large `syslog file`, so you may want to only use logging when you are troubleshooting.

Entries in `/var/adm/syslog/syslog.log` are marked with the monitor daemon name, for example `pkgmond` or `fsmond`, followed by the resource name and logging data. Additions, deletions, notifications, and changes in resource states are logged. Errors explaining why a resource is not available for monitoring, or why the monitor cannot access a resource are also logged there.

Look at the `registrar.log` if you are having trouble finding resources that you suspect exist on your system. This log contains any errors that were encountered when trying to read the dictionary. If a dictionary was corrupted in any way, the registrar would not be able to read it, and EMS would not be able to find the resources associated with that dictionary.

EMS Tracing

Some monitors provide tracing which can be used for debugging monitor code.

Use the `-d` option to turn on tracing for EMS. Tracing should only be used at the request of your HP support personnel when trying to determine if there may be a problem with EMS. To turn on tracing, modify the `.dict` file in `/etc/opt/resmon/dictionary` and add `-d` to the monitor you would like to trace:

```
MONITOR:  /etc/opt/resmon/lbin/mibmond -l -d
```

Kill the monitor process. The monitor will automatically restart with tracing enabled. To speed up monitor restart, use the `resls` command with the top level of the resource class as an argument, for example, `resls /system`.

Tracing is customarily logged to `/etc/opt/resmon/log/`*`monitor_name`*`.log`. The *`monitor_name`* usually matches the name used for the monitor in the dictionary file. For example, the MIBmonitor uses `mibmond.dict` and `mibmond.log`.

Performance Considerations

Monitoring your system is important to maintain high availability, but monitoring consumes system resources. You must carefully consider your performance needs against your need to know as soon as possible when a failure threatens availability.

System Performance Issues

The primary performance impact will be related to the polling interval and the number of resources being monitored. You need to balance your need to quickly detect failures with your need for system performance and adjust the number of resources you monitor and the polling intervals accordingly.

You may want to set a short interval, such as 30 seconds, for resources that require quick response after failure. You may want to set a longer polling interval, such as 5 minutes or more, for all other resources.

Network Performance Issues

Although monitoring is not likely to affect network performance, you may want to make sure that only necessary messages are being sent. Make sure your monitor requests are configured so you are notified only for important events.

Testing Monitor Requests

To test that events are being sent, use the INITIAL option available with conditional notification when creating a monitoring request. This option sends notification on startup. Examine it to make sure your request is properly configured and showing up in the correct system management tool.

An alternative is to use the “At each interval” notification to test that events are being sent in the correct system management tool. Once you establish that events are being sent properly, you can modify the request.

Testing Cluster Monitor Requests

Use the `cmviewcl -v` command to display detailed information about the current status of the cluster and packages on the cluster. The EMS cluster monitor should return the same values as this command.

Testing Network Monitor Requests

If you want to test whether events are sent in case of network failure, use the `/usr/bin/ifconfig` *LANname* `down` command to bring a card down, and examine the event to make sure it shows up in the correct system management tool.

Testing System Resource Monitor Requests

Use the `uptime` command to verify the number of users and the system load. The EMS system resource monitor should return the same values that this command does.

Making Sure Monitors are Running

Monitor daemons automatically start when you create a monitoring request. Because monitoring is designed to work in a high-availability environment, monitors are written to automatically restart if anything causes them to fail.

A daemon called `p_client` restarts all appropriate monitors using the monitor restart interval defined in `/etc/opt/resmon/config`. Therefore, a monitor cannot be permanently stopped or started by a

human.

Because the monitors are persistent, monitoring requests are kept when you install a new monitor or update an existing monitor. If a condition, such as “status > 3” is being monitored for a resources that has a range of 1-7, and new version of monitor is installed that supports a new status value, such as “8”, you may start seeing notifications for “status=8”.

MIB Monitor Troubleshooting

The MIB monitors that ship with EMS rely on various SNMP MIBs and need to have HP-UX SNMP subagents configured correctly and be running, before they can reliably report on the status of their resources. Other monitors that may be added might also need special SNMP configurations. Review the following troubleshooting hints to help ensure that your environment is set up correctly:

- Refer to the standard `/var/adm/syslog/syslog.log` file. It is always useful when troubleshooting system and ServiceGuard concerns.
- Certain log files may grow without bound. This may fill up file systems and cause unpredictable behavior in SNMP. Check (and possibly remove) the following files:

  ```
  /var/adm/snmpd.log
  $ORACLE_HOME/network/log/listener.log
  $ORACLE_HOME/rdbms/audit/*.aud
  $ORACLE_HOME/rdbms/log/*.trc
  ```

- If MIB resource classes under *`system`*, *`rdbms`*, *`cluster`*, and *`net`* are unavailable, there might be a problem with the HP SNMP daemons (`snmpdm`, `mib2agt`, and `hp_unixagt`). Try using the following commands to stop and restart HP SNMP:

  ```
  /sbin/init.d/SnmpMaster stop
  /usr/sbin/snmpd
  ```

NOTE On HP-UX version 10.20, if `trapdestagt` was running, it might need to be restarted manually with the command `/usr/sbin/trapdestagt`. NNM or OV depends on `trapdestagt` to set up SNMP trap notification on managed systems.

- If changes are needed to *`dictionary`* files, stop any MIB monitors that are already running.

  If changes are needed to `snmpd.conf`, stop and restart HP SNMP, following the procedure above.

NOTE The `-c` option is not required if `public` is one of the `get-community` or `set-community` names.

Glossary

A-H

alert An event. A message sent to tell a user or application when that certain conditions are met, an action or state you want to know about. For example, you may want to be alerted when a disk fails or when available filesystem space falls below a certain level.

asynchronous monitor A monitor that monitors resource instances (or resource class) asynchronously. It is event driven and send notifications when events occur. It does not keep track of the current state or value of each resource it monitors.

client The application that creates or cancels requests to monitor particular resources. The consumer of a resource status message. A user of the Resource Monitor framework. This user may browse resources, request status, and make requests to have resources monitored. Examples are MC/ServiceGuard as it starts a package or the SAM interface to EMS.

dictionary See Resource Dictionary.

Event Monitoring Service (EMS) A means to create requests, monitor, and report events about resources on a system. EMS observes a system and does not modify it.

EMS Framework A set of APIs together with the registrar process and the resource dictionary, which allows client applications to request that resources be monitored and a target application be notified.

EMS API The interface between the registrar, client applications, target applications, and resource monitors.

EMS GUI The SAM interface to EMS. One type of a client application, use it to create monitoring requests.

event An alert.

HA High Availability.

I-L

ITO HP OpenView IT/Operations, formerly known as Operations Center

logical extent The basic allocation unit for a logical volume is called a logical extent. For mirrored logical volumes, either two or three physical extents are mapped for each logical extent, depending on whether you are using 2-way or 3-way mirroring.

logical volume The segments of spaces that can be separated physically on a disk or be on serial disks. Each collection appears to the operating system as a single disk. Like disks, logical volumes can be used to hold file systems, raw data areas, dump areas, or swap areas. Unlike disks, logical volumes can be given a size when they are created, and a logical volume can later be expanded or reduced. Also, logical volumes can be spread over multiple disks.

LUN (Logical Unit Numbers) A logical disk device composed of one or more physical disk mechanisms, typically configured into a RAID level.

LVM (Logical Volume Manager) Software that manages disks in volume groups, and allows you to create logical and physical volume groupings.

M

MIB (Management Information Base). A document that describes objects to be managed. A MIB is created using a grammar defined in "Structure of Management Information" (SMI) format. This grammar concisely defines the objects being managed, the data types these objects take, descriptions of how the objects can be used, whether the objects are read-only or read-write, and assigns identifiers for the objects.

MIB II (MIB2) An MIB that *defines* information about the system, the network interface cards, routing information, the TCP and UDP sockets and their states, and various statistics related to error counts. This MIB is widely adopted and is served by most IP-addressed devices. Most system and network resources managed by EMS HA Monitors are taken from this MIB.

monitor See resource monitor.

N-P

notification See alert.

physical extent LVM divides each physical disk into addressed units called physical extents.

physical volume A disk that has been initialized by LVM.

polling The process by which a monitor obtains the most recent status of a resource. The method is defined by the monitor when it is created.

polling interval Determines the maximum amount of elapsed time before the monitor knows about a change in resource status.

protocol The method used to send notification messages. The options through EMS include: opcmsg, SNMP, TCP, UDP, syslog, console, textlog and email.

PVG (physical volume group)

A grouping of physical devices (host adapters, busses, controllers, or disks), that allow LVM to manage redundant links or mirrored disks and access the redundant hardware when the primary hardware fails.

PV links A method of LVM configuration that allows you to provide redundant SCSI interfaces and buses to disk arrays, thereby protecting against single points of failure in SCSI cards and cables.

Q-R

registrar Software that provides the link between clients (resource status consumers) and resource monitors (resource status providers). The central part of the resource monitor framework which uses the resource dictionary to act as an intermediary between client systems and resource monitors.

resource Any entity that a monitor application developer names. Examples include a network interface, CPU statistics, a MIB object, and a network service.

resource class A group of EMS resources organized into a filesystem type structure. Examples include system resources and cluster resources. See resource instance.

resource dictionary A set of files that provide to the registrar a hierarchy of resources on the local system and respective resource monitors.

resource instance The actual resources that can be monitored. For example, /net/interfaces/lan/status/lan0 may refer to a particular network interface installed on the monitored system. See resource class.

resource monitor The process that is used to obtain the status of a resource and send event notifications if appropriate. A monitor checks resources on the local system. The resource monitor maps the physical resource into a standard interface understood by EMS.

resource path A full resource path includes the resource class hierarchy and instance.

S-T

SNMP (Simple Network Management Protocol) Standard protocol for network-based retrieval of information about system resources.

state The current value of a resource (UP or DOWN). For some resource instances, a monitor may need to maintain a history of past events or conditions in order to know the resource value. In this case, a monitor is said to be maintaining state information. Stateless monitors do not keep any history of past conditions.

target The target application is notified when a monitored resource reaches the condition for which notification was requested. For example, a target application could be MC/ServiceGuard or IT/Operations (ITO).

U-Z

volume group In LVM, a set of physical volumes whose extents are grouped together and then made available to users as logical volumes. A volume group can be activated by only one node at a time unless you are using MC/LockManager. MC/ServiceGuard can activate a volume group when it starts a package. A given disk can belong to only one volume group. A logical volume can belong to only one volume group.

Index

Index