Chapter 20
-
Event Monitoring Services (EMS)

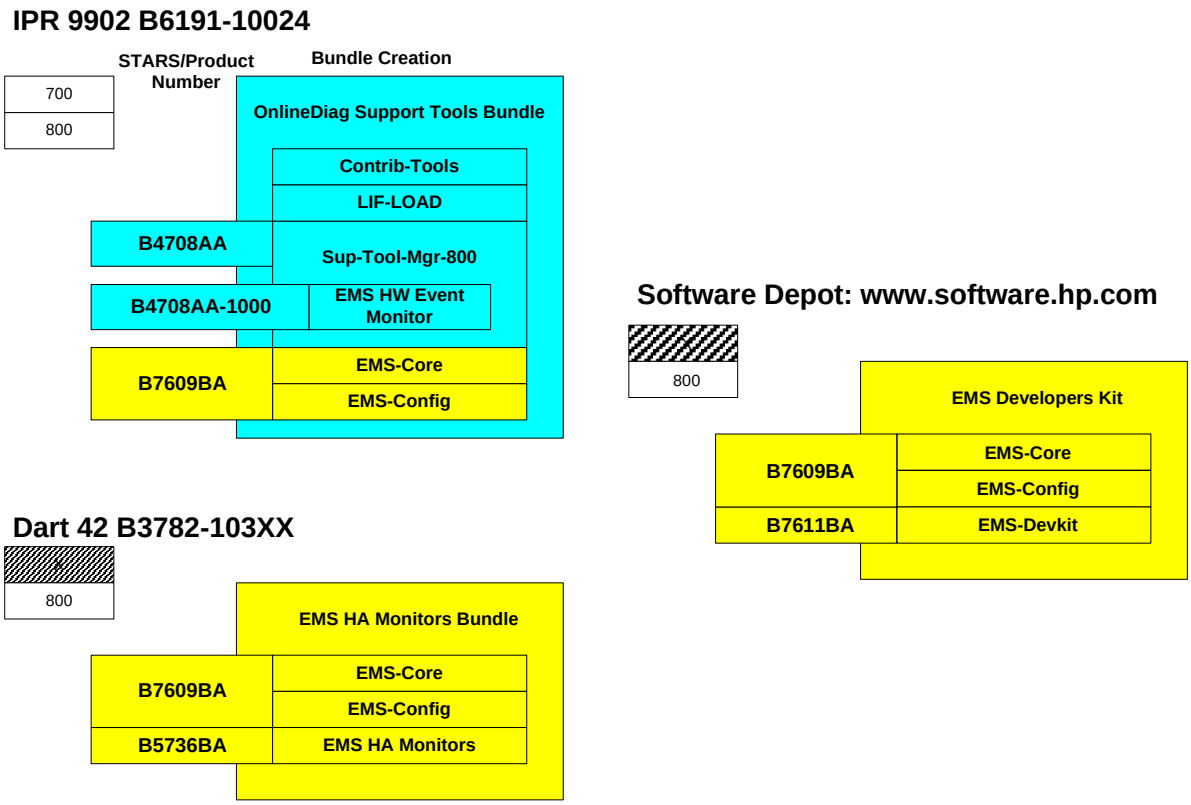
INDEX

Introduction

Since there has always been a little confusion about EMS i will first explain the different products and where to get them. Then I will describe the basic functionality and troubleshooting of EMS.

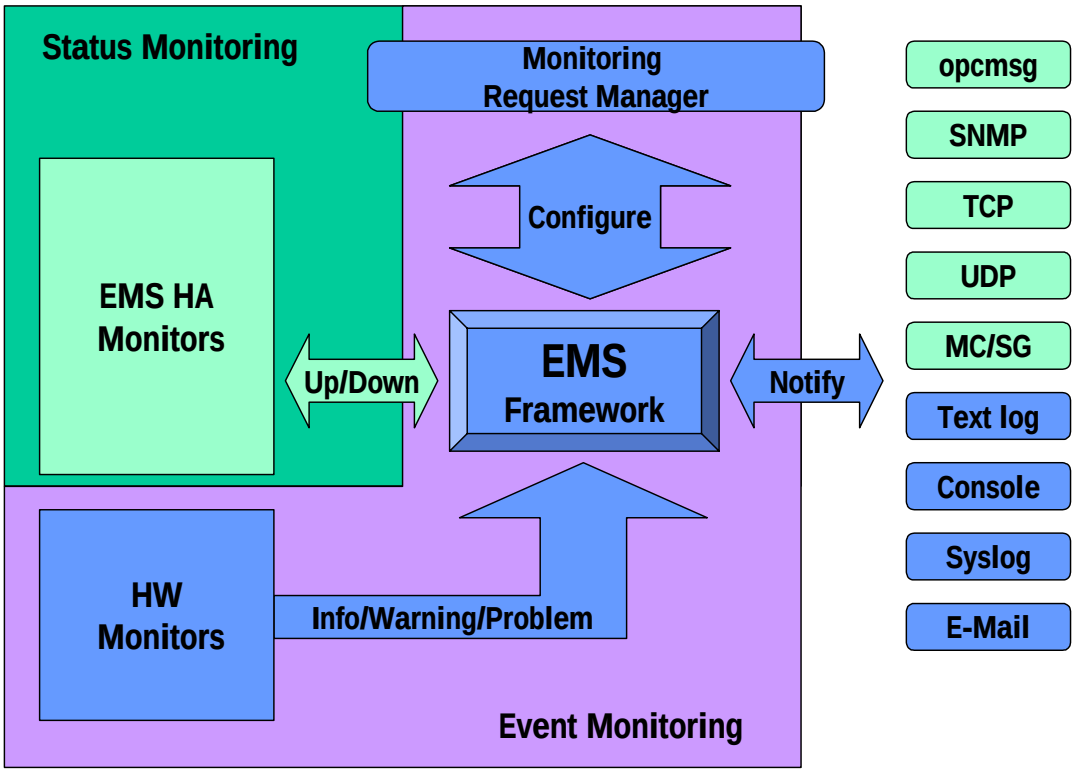
The Event Monitoring Services are a part of the Online Diagnostics. This part is also called HW-Monitoring. The HW-Monitors watch for the Hardware and report via different notification channels to the user or sysadmin.
The second product, which is also called EMS, are the HA-Monitors (High Availability). These Monitors use the same GUI as the HW-Monitors. Contrary to the HW-Monitors the HA-Monitors are not cost free. The product has to be ordered at extra charge. The HA-Monitors are supported by the software folks whereas the HW-Monitors are supported by the hardware folks.

The following overview shows the different products with their product numbers:



The EMS Architecture

EMS can be used to monitor resources with events and status. With status monitoring the monitor gives information such as up or down. With event monitoring the monitor brings a detailed message of the event.
The Notification is the same in both cases. You can use the same Notification for Event and Status Monitoring. The Event Monitoring is used by EMS Hardware. The Status Monitoring is used by Hardware and HA-Monitors.
Here is an Overview of the Architecture of EMS:



Installation and Configuration of EMS

EMS Hardware Monitoring is bundled with the Online Diagnostics but it can also be used without it. The HA Monitors are installed as additional Software Bundle. The graphical user interface and the configuration are the same. Remember that EMS HA Monitoring is only possible at extra cost for the customer.

The installation procedure is very simpe because you can use the product from every DART (Application) CD delivered to the customer. The following files come with the Online Diagnostics product:

- Files Installed - EMS Bundles:

```
EMS-Config.EMS-GUI,  EMS-Core.EMS_CORE
/opt/resmon/..
/etc/opt/resmon/..
/var/opt/resmon/..
```

- Startup scripts:

  Starts EMS persistence client: `/sbin/init.d/ems`
  based on config file: `/etc/rc.config.d/ems`
  Starts the EMS SNMP subagent: `/sbin/init.d/emsa`
  based on config file: `/etc/rc.config.d/emsagtconf`

- EMS Daemons started:

  At system boot:
  Persistence Client: `/etc/opt/resmon/lbin/p_client`
  EMS SNMP subagent: `/etc/opt/resmon/lbin/emsagent`
  At configuration time upon client connect:
  Registrar: `/etc/opt/resmon/lbin/registrar`

- System files modified:

```
/etc/inittab
ems1::bootwait:touch /etc/opt/resmon/persistence/reboot_flag
ems2::bootwait:/usr/bin/rm -f /etc/opt/resmon/persistence/runlevel4_flag
ems3::once:touch /etc/opt/resmon/persistence/runlevel4_flag
ems4:23456:respawn:/etc/opt/resmon/lbin/p_client

/etc/services
registrar      1712/tcp        # resource monitor service
registrar      1712/udp        # resource monitor service
```

swlist Output with HW monitors installed:

```
# swlist -l product | grep EMS
  EMS-Config            A.03.20        EMS Config
  EMS-Core              A.03.20        EMS Core Product
  EMS-KRMonitor         A.11.00.01     EMS Kernel Resource Monitor
  EMS-RdbmsMon          A.03.20        EMS Database Monitor
```

swlist output with HA monitor installed additionally:

```
# swlist -l product | grep EMS
  EMS-Config            A.03.20        EMS Config
  EMS-Core              A.03.20        EMS Core Product
  EMS-DiskMonitor       A.03.20        EMS Disk Resource Monitor
  EMS-KRMonitor         A.11.00.01     EMS Kernel Resource Monitor
  EMS-MIBMonitor        A.03.20        EMS MIB Resource Monitor Product
  EMS-RdbmsMon          A.03.20        EMS Database Monitor
```

To verify the product is functional:

Execute the "`resls /`" command to display any EMS resource.
Run the EMS GUI and add a monitoring request.

Configuration Hardware Monitor

The Hardware monitors are divided into Event and Status monitoring.
The Event monitoring can be configured with the `monconfig` program. This tool is stored in `/etc/opt/resmon/lbin` and has only a character interface. After installation of Event monitoring there are some monitors already configured.

The following list shows the default configured monitors from Hardware Event Monitoring:

- Defaults in IPR 9902

Send events generated by all monitors with severity >= INFORMATION to TEXTLOG (`/var/opt/resmon/log/event.log`)
Send events generated by all monitors with severity >= SERIOUS to SYSLOG
Send events generated by all monitors with severity >= SERIOUS to CONSOLE
Send events generated by all monitors with severity >= SERIOUS to EMAIL root

- Defaults changed in IPR 9904

Events with a severity >= MAJOR WARNING to SYSLOG, CONSOLE and to EMAIL root

- Predictive adds an additional default request for FC monitors

Predictive configures disk_em, dm_ses_enclosure, dm_FCMS_adapter, and dm_fc_scsi_mux monitors to write events with a severity >= INFORMATION to TEXTLOG (`/var/opt/pred/emslog`)

monconfig shows the actual state of Event Monitoring. In this examples Monitoring is disabled:

```
================================================================================
==================       Event Monitoring Service       ==================
==================      Monitoring Request Manager      ==================
================================================================================

         EVENT MONITORING IS CURRENTLY DISABLED.

================================================================================
==============      Monitoring Request Manager Main Menu      ==============
================================================================================

Select:
   (S)how monitoring requests configured via monconfig
   (C)heck detailed monitoring status
   (L)ist descriptions of available monitors
   (A)dd a monitoring request
   (D)elete a monitoring request
   (M)odify an existing monitoring request
   (E)nable Monitoring
   (K)ill (disable) monitoring
   (H)elp
   (Q)uit
   Enter selection: [s]
```

With monconfig we can add or delete monitoring request. In the following section we add one monitor request:

```
================================================================================
====================      Add Monitoring Request      ====================
================================================================================
Start of edit configuration:

A monitoring request consists of:
  - A list of monitors to which it applies
  - A severity range (A relational expression and a severity.  For example,
    < "MAJOR WARNING" means events with severity "INFORMATION" and
    "MINOR WARNING")
  - A notification mechanism.
Please answer the following questions to specify a monitoring request.

Monitors to which this configuration can apply:
   1) /storage/events/disk_arrays/AutoRAID
   2) /storage/events/disks/default
   3) /adapters/events/FC_adapter
   4) /connectivity/events/multiplexors/FC_SCSI_mux
   5) /system/events/memory
   6) /storage/events/enclosures/ses_enclosure
   7) /storage/events/tapes/SCSI_tape
   8) /storage/events/disk_arrays/FW_SCSI
   9) /storage/events/disk_arrays/High_Availability
   Enter monitor numbers separated by commas
   {or (A)ll monitors, (Q)uit, (H)elp} [2]
```

The monitors are arranged into classes and subclasses. This is similar as a directory structure on a disk. So in this example we take a disk monitor /storage/events/disks/default
After selection of the monitor we need to configure the criteria when the monitor will notify me. There are 5 levels.

```
Criteria Thresholds:
   1) INFORMATION    2) MINOR WARNING    3) MAJOR WARNING
   4) SERIOUS        5) CRITICAL
   Enter selection {or (Q)uit,(H)elp} [4]
```

The default value is (4) . The next step is to setup the operator for the notification. Here we need to decide the level which is also be notified by the monitor.

```
Criteria Operator:
   1) <      2) <=      3) >      4) >=      5) =      6) !=
   Enter selection {or (Q)uit,(H)elp} [4]
```

Here is (4) also the default, which means that we monitor all event greater or equal SERIOUS.

The last step is to setup the notification for the monitor.

```
Notification Method:
   1) UDP        2) TCP        3) SNMP       4) TEXTLOG
   5) SYSLOG     6) EMAIL      7) CONSOLE    8) OPC
   Enter selection {or (Q)uit,(H)elp} [6]
```

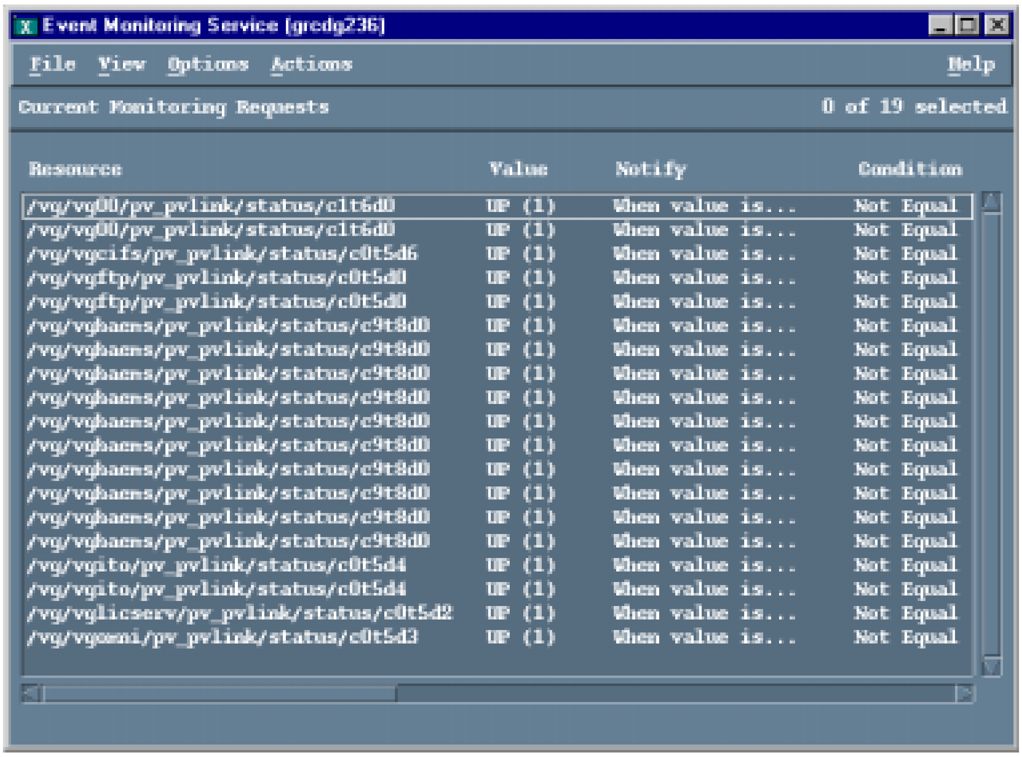
Here we can setup where the monitor will send his output.

Now we have added one monitor. The request will be stored in the `/etc/opt/resmon/persistence` directory as a file named `m.xxxx`. The value for `xxx` depends on the monitor. Every monitor has a unique number. These files are read from the p_client process which setup the request.

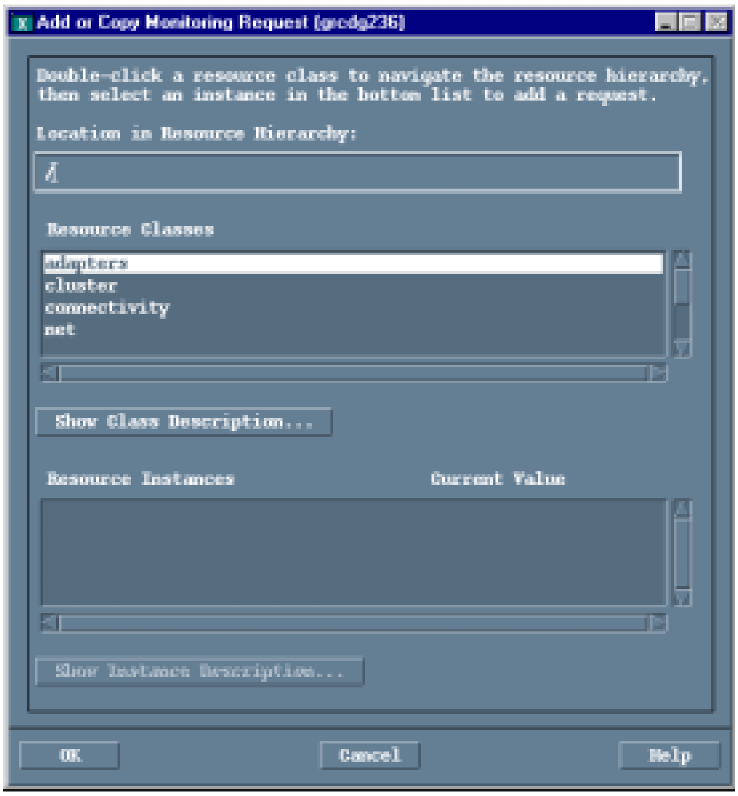
The GUI to configure Hardware Monitor request is used with SAM. When SAM is started There is a entry Resource Management. This entry shows a new submenu with an entry like Event Monitoring Service. This will start the GUI interface.

NOTE: With the GUI Interface we can only set up the Status Monitoring for Hardware and HA Monitors. The HA monitors didn't come with Event Monitoring.

Here is a snapshot of the GUI:

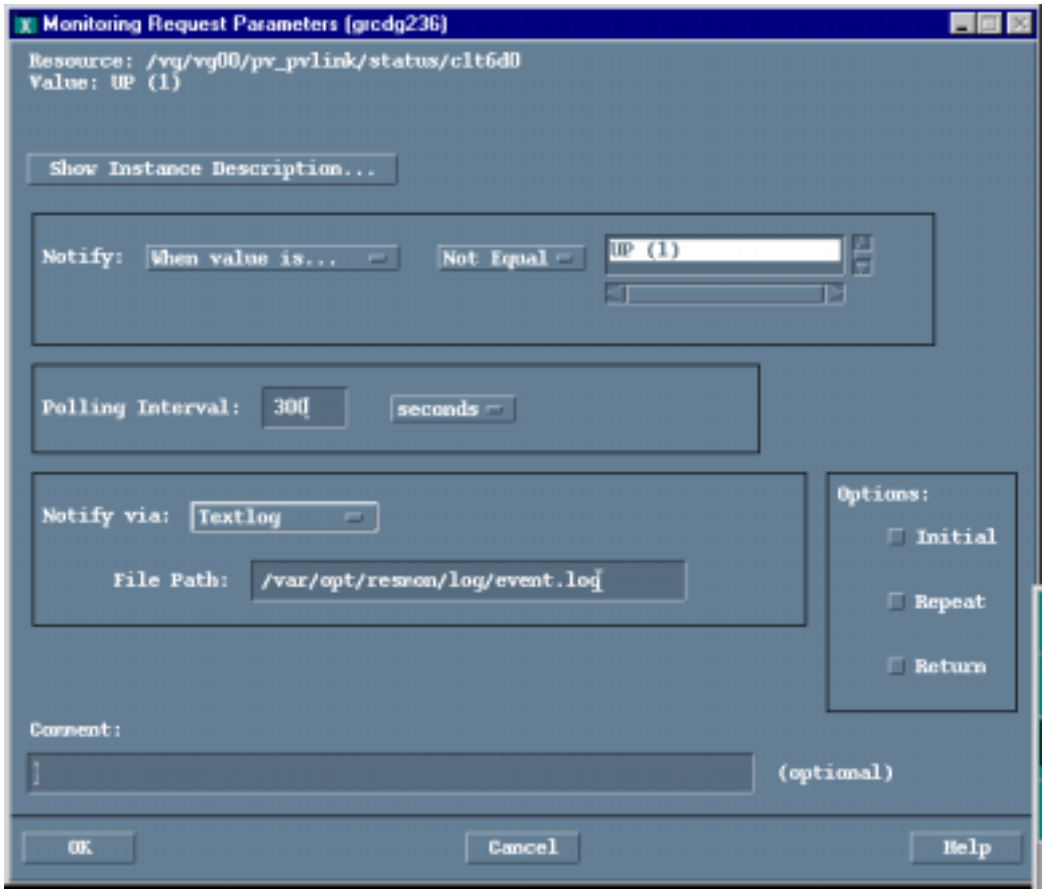


The Status of a monitor is often up/down. Some other values are unknown or suspect. To configure additional requests you need to klick *Action* and add new Monitor request. Then the following box comes up:



From this Window we can select the Monitor. Monitors are combined into classes. To open a subclass click on the desired class. Every class contains differen subclasses and monitos. For examples the class /vg contains a subclass like <vgname>/pv_pvlink. This will monitor the pvlink of a specified disk (this is an HA monitor).
The class storage contains a subclass /storage/status/disk. This list all devices and then the device can be monitored. (This is a HW monitor).
In the Appendix there are links to get documents for HA and HW monitoring description.

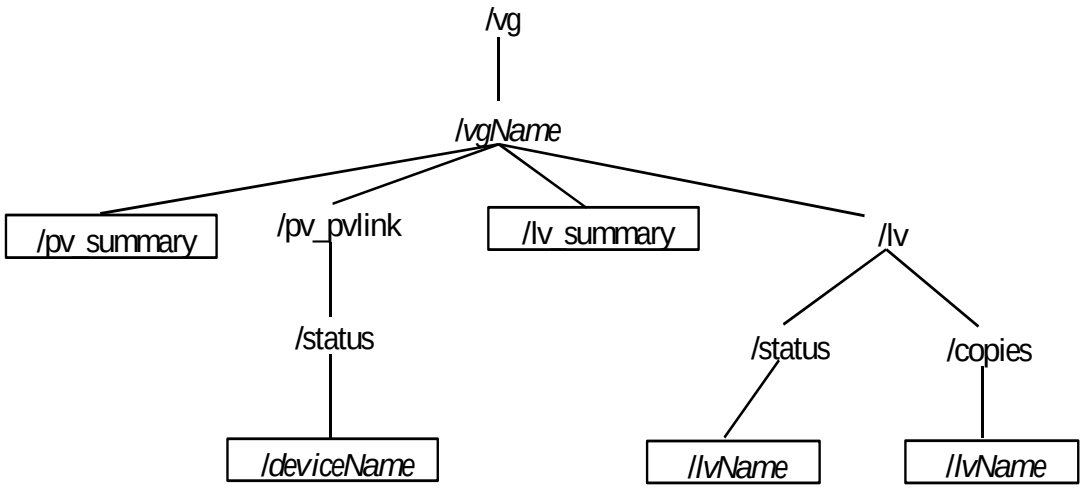
In this example we take the following Monitor to add `/vg/vg00/pv_pvlink/status/c1t6d0`



Here we can set up the polling interval and notification for this monitor.

NOTE: If you are using the text interface from SAM there is a problem to open subclasses.
If you stay for example on the class /vg and own this class. The class will not open with a <CR>. In a hpterm emulation you need to type the function key F2 and then Enter. Now the class will open. In a xterm window you need the funtion Key F1.
If you change the Notification for a Monitor that is equal UP, you must stay on the Show Instance Description item and type Funtion Key F2 and the Tabulator key. Then you can access the changed window. Otherwise you will move with a Tab Key to Polling interval.
Now we have configured Monitor requests for HA and Hardware Monitoring.

Here is an example of the HA monitor /vg with his subclasses:



Overview HA and HW Monitors

This chapter is an overview which Resource is a Hardware or HA Monitor

HA Monitors

- *Disk Monitor*

  Monitors: LVM Disk status
  Resources: `/vg/…`

- *System Monitor*

  Monitors: Number of Users, Job Queue Size, and File System Free Space
  Resources:
  ```
  /system/numUsers
  /system/jobQueue1Min
  /system/jobQueue5Min
  /system/jobQueue15Min
  /system/filesystem/availMb
  ```

- *Network Monitor*

  Monitors: LAN interface status
  Resources: `/net/interfaces/lan/status`

- *Cluster Monitor*

  Monitors: Cluster, Node, Package, Service status
  Resources:
  ```
  /cluster/status
  /cluster/package/package_status
  /cluster/package/service_status
  ```

HW Monitors

- *AutoRAID Disk Array Monitor*

  Monitors: Model 12H Disks status
  Resources: `/storage/status/disk_arrays/AutoRAID`

- *High Availability Disk Array Monitor*

  Monitors: NIKE Model 10 and 20 SCSI Diskarrays,
  NIKE Model 30/FC Diskarray status
  Resources: `/storage/status/disk_arrays/High_Availability`

- *Disk Monitor*

  Monitors: SCSI and fibre channel disks (not disk arrays)
  Resources: `/storage/status/disks/disk_em`

- *SCSI Tape Device Monitor*

  Monitors: HP-PB and HSC tape status
  Resources: `/storage/status/tapes/SCSI_tape`

- *High Availability Storage System Monitor*

  Monitors: Fibre Channel storage enclosures (not disks within enclosure)
  Resources: `/storage/status/enclosures/ses_enclosure`

- *Fast/Wide SCSI Disk Array Monitor*

  Monitors: Models C243XHA status
  Resources: `/storage/status/disk_arrays/FW_SCSI`

- *Fibre Channel SCSI Multiplexer Monitor*

  Monitors: Model A3308A status
  Resources: `/connectivity/status/multiplexors/FC_SCSI_mux`

- *Fibre Channel Adapter Monitor*

  Monitors: FC Adapters Models A3404A, A3636A, A3591A, A3740A status
  Resources: `/adapters/status/FC_adapter`

- *Fibre Channel Arbitrated Loop Hub Monitor*

  Monitors: Models A3724A, A4839A status
  Resources: `/connectivity/status/hubs/FC_hub`

- *Memory Monitor*

  Monitors: all system memory status
  Resources: `/system/status/memory`

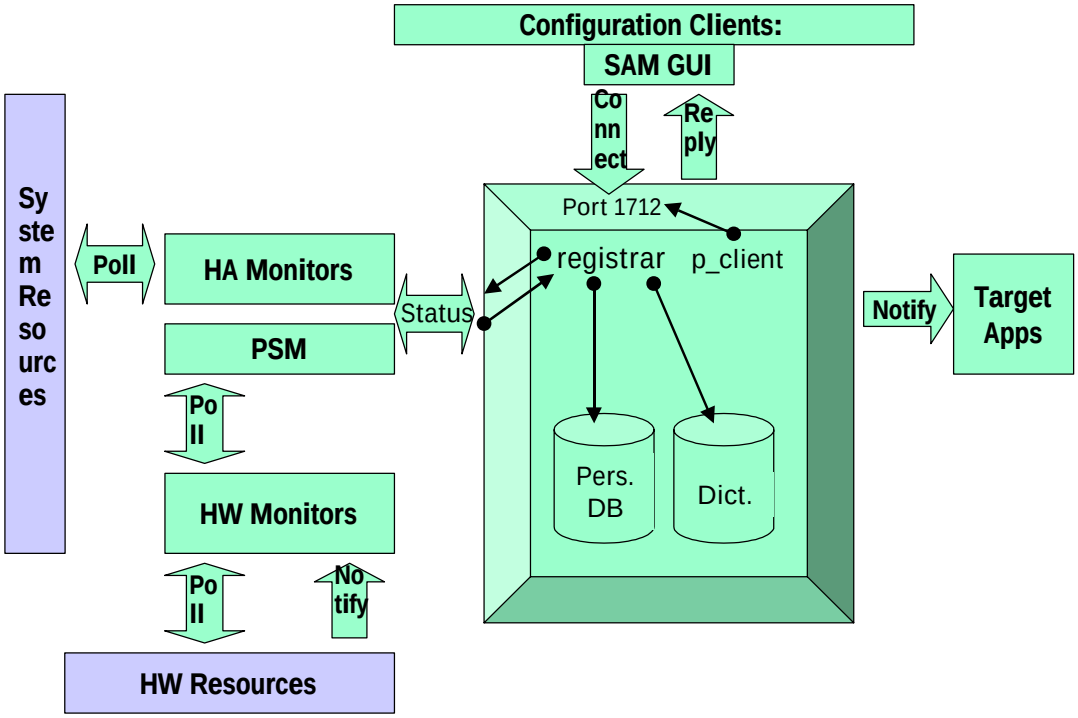
Overview:

Resource	(Type) Monitor	Requirement
`/vg/...`	(HA) diskmond	scsi pass-thru driver spt0
`/net/...`	(HA) lanmond	/usr/sbin/mib2agt (*)
`/system/numUsers`	(HA) mibmond	/usr/sbin/hp_unixagt (*)
`/system/jobQueue*Min`	(HA) mibmond	/usr/sbin/hp_unixagt (*)
`/system/filesystem/...`	(HA) fsmond	/usr/sbin/hp_unixagt (*)

`/cluster/...`	(HA) clustermond	/usr/sbin/cmsnmpd (*)
`/cluster/package/...`	(HA) pkgmond	/usr/sbin/cmsnmpd (*)
`/cluster/package/service/...`	(HA) svcmond	/usr/sbin/cmsnmpd (*)
`/rdbms/...`	(HA) rdbmsmond	dbsnmp, ora_naaagt, master_peer
`/storage/status/...`	(HW)	>>> HW Configuration
`/connectivity/status/...`	(HW)	>>> HW Configuration
`/adapters/status/...`	(HW)	>>> HW Configuration
`/system/status/memory`	(HW)	>>> HW Configuration

The Persistence Client p_client , Registrar

This picture shows the complete EMS enviroment with the used daemons:



Once a monitor request has been accepted by a monitor, the request information is stored in the persistence database (files under the directory /etc/opt/resmon/persistence).

Persistence files are based on a monitors named pipe file name.

```
# ls /etc/opt/resmon/persistence
m.1445432762    m.169145941     m.2765141061    m.3853054343
m.1533300606    m.1977287795    m.3088511186    m.4077902835
#
```

p_client (persistence client) is started by init (1M), and respawned by init if it dies.

p_client checks for dead monitors and restarts them (and any outstanding monitoring

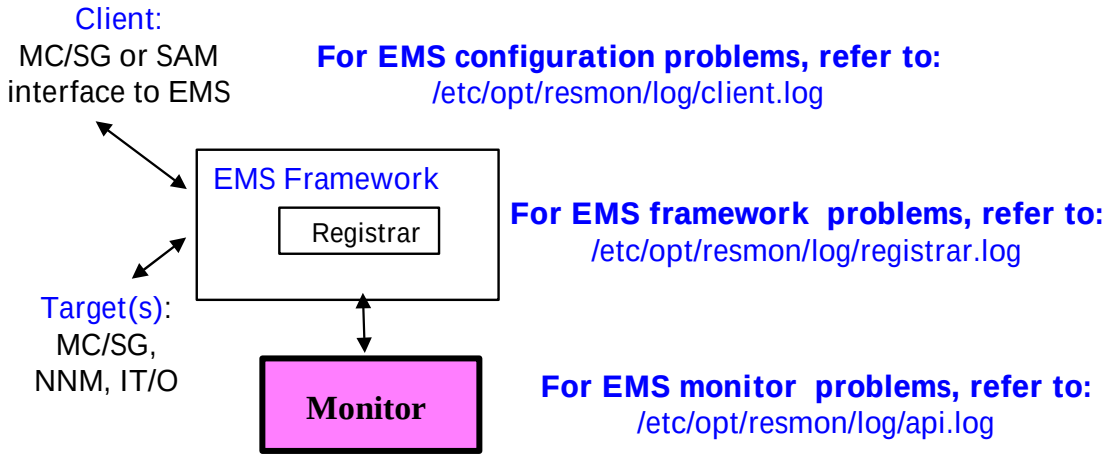
requests).

Default interval is every 2 minutes (interval set in `/etc/opt/resmon/config`)

p_client runs at system startup to restart monitors and any outstanding monitoring requests

EMS Logging

There are three interesting files where EMS logs all information:



This three files are the most interesting files from EMS. But there is also the default log file `/var/opt/resmon/event.log`. This file will be used for HW Monitoring and default Notification if Text file is used.

The three logfiles will grow and when they reached about 500kB then they are automatically moved to `<logfile>.old`. If the logfile is increased to 500kB it will be moved again. So the `/etc/opt/resmon/log` directory cannot run in full filesystem.

EMS DEBUG Mode

To get more information for troubleshooting create a file `/etc/opt/resmon/debug`. If this file exists EMS log everything in Debug Level. But remember that the file size is also set to 500 kB.

To log resource state changes for EMS HA Monitors, use the "`-l`" option.
To debug EMS HA monitors, use the "`-d`" option to turn on tracing.
Modify `/etc/opt/resmon/dictionary/<monitor>.dict` file.

For example:

Disk Monitor dictionary file "diskmond.dict":

```
/etc/opt/resmon/lbin/diskmond -l –d
```

State changes logged to:

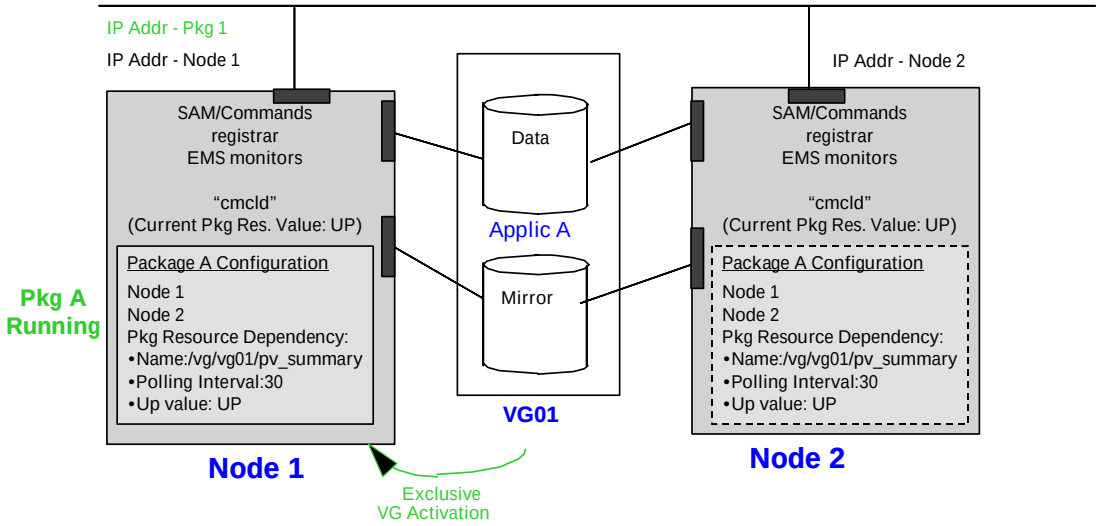
```
/var/adm/syslog/syslog.log
```

Tracing logged to:

```
/etc/opt/resmon/log/diskmond.log
```

EMS with MC/ServiceGuard

EMS monitos can be used to configure Package dependencies into MC/ServiceGuard. Every dependency must exist on each node. It is possible to add Hardware or HA Monitors into ServiceGuard Scripts.
The Monitor can be added into the .conf Script of the Package. If you add new Monitors to the Cluster the cluster must be down. Changing of already configured Monitors in the cluster is possible.



If the state of a package dependency goes DOWN, the package is started on an alternate node where the resource is known to be in an UP state.

Create a package dependency

Performed from the Clusters => HA Clusters area of SAM
Perform the action *Create/Add a Package* or *Modify Package Configuration*
Specify Package Resource Dependencies
Add desired monitored resources to package dependencies
The *Add Resource* dialog is identical to the EMS-GUI dialog
Enter Resource parameters
Package Dependencies can also be created by editing the package configuration file in `/etc/cmcluster/pkg.ascii`. Modify the variables:

```
RESOURCE_NAME        <Full_path_name>
RESOURCE_POLLING_INTERVAL       <numeric_seconds>
RESOURCE_UP_VALUE       <op><string_or_numeric> [and <op> <numeric>]
```

Additional information

EMS User's Guide and Release Notes
http://docs.hp.com/hpux/ha

EMS Developer's Kit
http://www.software.hp.com/products/EMS

Lab website
http://haweb.cup.hp.com/EMS (HP internal)

DST Sentinel web page with R&D Lab information, monitor ERS's, white papers
http://wojo.rose.hp.com/Sentinel_top.htm (HP internal)

EMS Training Material
http://haweb.cup.hp.com/ATC/Training/EMS/overview.html (HP internal)