Managing Faults in Oracle® Solaris 11.2

Part No: E48546
July 2014

ORACLE

Contents

Using This Documentation

- **Overview** – Describes how to use the Oracle Solaris Fault Management Architecture (FMA) feature to manage hardware faults and some software defects. FMA is one of the components of the wider Oracle Solaris Predictive Self Healing capability.
- **Audience** – System administrators who monitor and handle system faults and defects
- **Required knowledge** – Experience administering Oracle Solaris systems

Product Documentation Library

Late-breaking information and known issues for this product are included in the documentation library at http://www.oracle.com/pls/topic/lookup?ctx=E36784.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPTER 1

Introduction to the Fault Manager

The Oracle Solaris OS includes an architecture for building and deploying systems and services that are capable of predictive self healing. The service that is the core of the Fault Management Architecture (FMA) receives data related to hardware and software errors, automatically diagnoses the underlying problem, and responds by attempting to take faulty components offline.

This chapter discusses the following topics:

- Description of the Oracle Solaris Fault Management feature
- Configuring when and how you will be notified of error events
- Features of messages from the Fault Manager

When specific hardware faults occur, Oracle Auto Service Request (ASR) can automatically open an Oracle service request. See http://www.oracle.com/asr for more information about ASR.

Fault Management Overview

The Oracle Solaris Fault Management feature includes the following components:

- An architecture for building resilient error handlers
- Structured error telemetry
- Automated diagnostic software
- Response agents
- Structured messaging

Many parts of the software stack participate in fault management, including the CPU, memory and I/O subsystems, Oracle Solaris ZFS, and many device drivers.

FMA can help with both faults and defects:

- Faults – A faulted component is a component that used to work but no longer works.
- Defects – A defective component is a component that never worked correctly.

Hardware can experience both faults and defects. Most software problems are defects or are caused by configuration issues. Fault management and system services often interact. For example, a hardware problem might cause services to be stopped or restarted. An SMF service error might cause FMA to report a defect.

The fault management stack includes error detectors, a diagnosis engine, and response agents.

Error detectors — Error detectors detect errors in the system and perform any immediate, required handling. Error detectors issue well-defined error reports, or *ereports*, to a diagnosis engine.

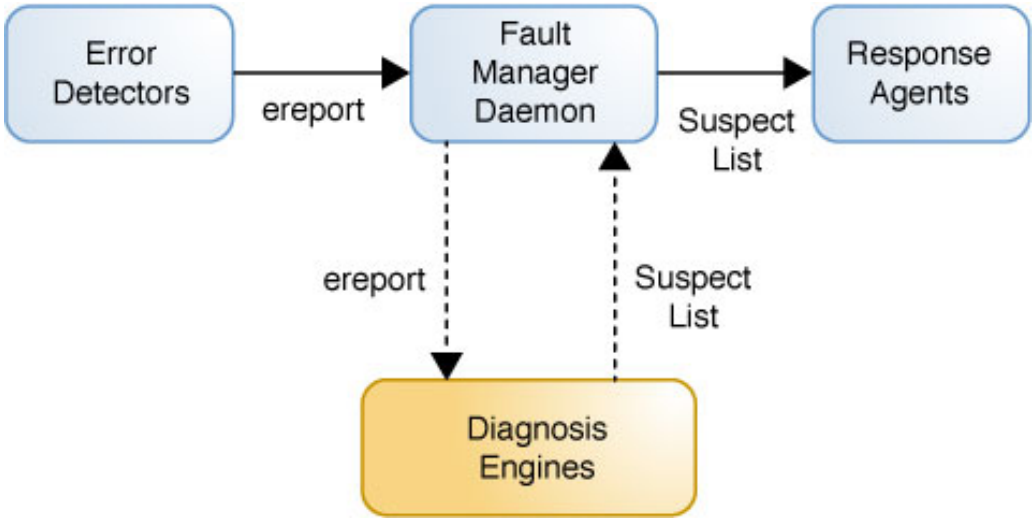
Diagnosis engine — The diagnosis engine interprets ereports and determines whether a fault or defect is present in the system. When such a determination is made, the diagnosis engine issues a *suspect list* that describes the resource or set of resources that might be the cause of the problem. The resource might have an associated Field Replaceable Unit (FRU), a label, or an Automatic System Reconfiguration Unit (ASRU). An ASRU might be immediately removed from service to mitigate the problem until the FRU is replaced.

When the suspect list includes multiple suspects (for example, if the diagnosis engine cannot isolate a single suspect), each suspect is assigned a probability of being the key suspect. The probabilities in this list sum to 100 percent. Suspect lists are interpreted by response agents.

Response agents — Response agents attempt to take action based on the suspect list. Responses include logging messages, taking CPU strands offline, retiring memory pages, and retiring I/O devices.

Error detectors, diagnosis engines, and response agents are connected by the Fault Manager daemon, `fmd`, which acts as a multiplexor between the various components, as shown in the following figure.

FIGURE 1-1 Fault Management Architecture Components



The lifecycle of a problem managed by the Fault Manager can include the following stages:

Diagnose	A new problem has been diagnosed by the Fault Manager. The diagnosis includes a list of one or more suspects. A suspect might have been automatically isolated to prevent further errors from occurring. The problem is identified by a UUID in the event payload, and further events describing the resolution lifecycle of this problem quote a matching UUID.
Update	One or more of the suspect resources in a problem diagnosis has been repaired, replaced, or acquitted, or the resource has faulted again. The suspect list still contains at least one faulted resource. A repair might have been made by executing an `fmadm` command, or the system might have detected a repair such as a changed serial number for a part. The `fmadm` command is described in Chapter 3, “Repairing Faults”.
Repair	All of the suspect resources in a problem diagnosis have been repaired, resolved, or acquitted. Some or all of the resources might still be isolated.
Resolve	All of the suspect resources in a problem diagnosis have been repaired, resolved, or acquitted and are no longer isolated. For example, a CPU that was a suspect and was offlined is now back online again. Offlining and onlining resources is usually automatic.

The Fault Manager daemon is a Service Management Facility (SMF) service. The `svc:/system/fmd` service is enabled by default. See "Managing System Services in Oracle Solaris 11.2 " for more information about SMF services. See the `fmd`(1M) man page for more information about the Fault Manager daemon.

The `fmadm config` command shows the name, description, and status of each module in the Fault Manager. These modules diagnose and repair problems on the system. The `fmstat` command displays additional information about these modules, as shown in "Fault Statistics" on page 20.

Receiving Notification of Faults and Defects

The Fault Manager daemon notifies you that a fault or defect has been detected and diagnosed.

Configuring When and How You Will Be Notified

Use the `svcs -n` and `svccfg listnotify` commands to show event notification parameters, as shown in "Showing Event Notification Parameters" in "Managing System Services in Oracle Solaris 11.2 ".

Use the `svccfg setnotify` command to configure FMA error event notification, as shown in "Configuring Notification of State Transition and FMA Events" in "Managing System Services in Oracle Solaris 11.2 ". For example, the following command creates a notification that sends an SMTP message when an FMA-managed problem is repaired:

```
$ svccfg setnotify problem-repaired smtp:
```

You can configure notification of fault management error events to use the Simple Mail Transfer Protocol (SMTP) or the Simple Network Management Protocol (SNMP).

FMA event tags include `problem-diagnosed`, `problem-updated`, `problem-repaired`, and `problem-resolved`. These tags correspond to the problem lifecycle stages described in "Fault Management Overview" on page 7.

Event notification and FMA event tags are also described in the "Notification Parameters" section in the `smf`(5) man page. For more information about the notification daemons, see the `snmp-notify`(1M) and `smtp-notify`(1M) man pages.

Settings for notification parameters for FMA events are stored in `fmnotify` properties in `svc:/system/fm/notify-params:default`. Events generated by SMF state transitions are stored in

the service or in the transitioning instance. System-wide notification parameters for SMF state transition events are stored in `svc:/system/svc/global:default`.

Understanding Messages From the Fault Manager Daemon

The Fault Manager daemon sends messages to both the console and the `/var/adm/messages` file. Messages from the Fault Manager daemon use the format shown in the following example:

```
1    SUNW-MSG-ID: SUN4V-8001-8H, TYPE: Fault, VER: 1, SEVERITY: Minor
2    EVENT-TIME: Wed Aug 24 21:56:03 UTC 2011
3    PLATFORM: SUNW,T5440, CSN: -, HOSTNAME: bur419-61
4    SOURCE: cpumem-diagnosis, REV: 1.7
5    EVENT-ID: 7b83c87c-78f6-6a8e-fa2b-d0cf16834049
6    DESC: The number of integer register errors associated with this thread has
7    exceeded acceptable levels.
8    AUTO-RESPONSE: The fault manager will attempt to remove the affected thread
9    from service.
10   IMPACT: System performance may be affected.
11   REC-ACTION: Use 'fmadm faulty' to provide a more detailed view of this
12   event. Please refer to the associated reference document at
13   http://support.oracle.com/msg/SUN4V-8001-8H for the latest service procedures and
14   policies regarding this diagnosis.
```

When you are notified of a diagnosed problem, consult the recommended knowledge article for additional details. Line 13 shows the recommended knowledge article in this example. The knowledge article might contain actions that you or a service provider should take in addition to other actions listed in the “REC-ACTION” section.

♦ ♦ ♦ CHAPTER 2

Displaying Fault Information

This chapter shows how to display detailed information about problems detected by the fault management system.

Displaying Information About Faults or Defects

Use the `fmadm faulty` command to display fault or defect information and determine which FRUs are involved. The `fmadm faulty` command displays active problems. The `fmdump` command displays the contents of log files associated with the Fault Manager daemon and is more useful as a historical log of problems on the system.

Tip - Base your administrative action on output from the `fmadm faulty` command. Log files output by the `fmdump` command can contain error statements that are not faults or defects.

The `fmadm faulty` command displays status information for resources that the Fault Manager identifies as faulty. The `fmadm faulty` command has many options for displaying different information or displaying information in different formats. See the `fmadm`(1M) man page for information about all the `fmadm faulty` options.

EXAMPLE 2-1 `fmadm faulty` Output Showing One Faulty CPU

```
# fmadm faulty
--------------- ------------------------------------  -------------- ---------
TIME            EVENT-ID                              MSG-ID         SEVERITY
--------------- ------------------------------------  -------------- ---------
Aug 24 17:56:03 7b83c87c-78f6-6a8e-fa2b-d0cf16834049  SUN4V-8001-8H  Minor

Host        : bur419-61
Platform    : SUNW,T5440        Chassis_id  : BEL07524BN
Product_sn  : BEL07524BN

Fault class : fault.cpu.ultraSPARC-T2plus.ireg
Affects     : cpu:///cpuid=0/serial=1F95806CD1421929
```

```
13                   faulted and taken out of service
14   FRU         : "MB/CPU0" (hc://:product-id=SUNW,T5440:server-id=bur419-61:\
15                 serial=3529:part=541255304/motherboard=0/cpuboard=0)
16                   faulty
17   Serial ID.  : 3529
18                 1F95806CD1421929
19
20   Description : The number of integer register errors associated with this thread
21                 has exceeded acceptable levels.
22
23   Response    : The fault manager will attempt to remove the affected thread from
24                 service.
25
26   Impact      : System performance may be affected.
27
28   Action      : Use 'fmadm faulty' to provide a more detailed view of this event.
29                 Please refer to the associated reference document at
30                 http://support.oracle.com/msg/SUN4V-8001-8H for the latest service
31                 procedures and policies regarding this diagnosis.
```

Line 14 identifies the impacted FRU. The string shown in quotation marks, “MB/CPU0,” should match the label on the physical hardware. The string shown in parentheses is the Fault Management Resource Identifier (FMRI) for the FRU. The FMRI includes descriptive properties about the system that contains the fault, such as its host name and chassis serial number. On some platforms, the part number and serial number of the FRU are also included in the FMRI of the FRU.

The Affects lines (lines 12 and 13) indicate the components that are affected by the fault and their relative state. In this example, a single CPU strand is affected. That CPU strand is faulted and has been taken out of service by the Fault Manager.

Following the FRU description in the `fmadm faulty` command output, line 16 shows the state as `faulty`. The Action section might include specific actions in addition to references to documents on the support site.

EXAMPLE 2-2 `fmadm faulty` Output Showing Multiple Faults

```
1    # fmadm faulty
2    --------------- ------------------------------------  -------------- -------
3    TIME            EVENT-ID                              MSG-ID         SEVERITY
4    --------------- ------------------------------------  -------------- -------
5    Sep 21 10:01:36 d482f935-5c8f-e9ab-9f25-d0aaafec1e6c  PCIEX-8000-5Y  Major
6
7    Fault class  : fault.io.pci.device-invreq
8    Affects      : dev:///pci@0,0/pci1022,7458@11/pci1000,3060@0
9                   dev:///pci@0,0/pci1022,7458@11/pci1000,3060@1
10                   ok and in service
11                  dev:///pci@0,0/pci1022,7458@11/pci1000,3060@2
12                  dev:///pci@0,0/pci1022,7458@11/pci1000,3060@3
13                    faulty and taken out of service
14   FRU          : "SLOT 2" (hc://.../pciexrc=3/pciexbus=4/pciexdev=0)
15                    repair attempted
16                  "SLOT 3" (hc://.../pciexrc=3/pciexbus=4/pciexdev=1)
```

```
17                  acquitted
18                "SLOT 4" (hc://.../pciexrc=3/pciexbus=4/pciexdev=2)
19                  not present
20                "SLOT 5" (hc://.../pciexrc=3/pciexbus=4/pciexdev=3)
21                  faulty
22
23    Description  : The transmitting device sent an invalid request.
24
25    Response     : One or more device instances may be disabled
26
27    Impact       : Possible loss of services provided by the device instances
28                   associated with this fault
29
30    Action       : Use 'fmadm faulty' to provide a more detailed view of this event.
31                   Please refer to the associated reference document at
32                   http://support.oracle.com/msg/PCIEX-8000-5Y for the latest service
33                   procedures and policies regarding this diagnosis.
```

In this output, device 1 in slot 3 is described as “ok and in service” on line 10, and line 17 shows its state as “acquitted.” Device 3 in slot 5 is described as “faulty and taken out of service,” and its state is “faulty.” States shown for two other devices are “repair attempted” and “not present.”

EXAMPLE 2-3 Showing Faults With the `fmdump` Command

Some console messages and knowledge articles might instruct you to use the `fmdump -v -u` *UUID* command to display fault information, as shown in the following example:

```
1    # fmdump -v -u 7b83c87c-78f6-6a8e-fa2b-d0cf16834049
2    TIME                 UUID                                 SUNW-MSG-ID EVENT
3    Aug 24 17:56:03.4596 7b83c87c-78f6-6a8e-fa2b-d0cf16834049 SUN4V-8001-8H Diagnosed
4      100%  fault.cpu.ultraSPARC-T2plus.ireg
5
6            Problem in: -
7               Affects: cpu:///cpuid=0/serial=1F95806CD1421929
8                   FRU: hc://:product-id=SUNW,T5440:server-id=bur419-61:\
9                   serial=9999:part=541255304/motherboard=0/cpuboard=0
10              Location: MB/CPU0
```

The information about the affected FRUs is on lines 8 through 10. The Location string on line 10 presents the human-readable FRU string. Line 8 shows the FMRI of the FRU. To see the severity, descriptive text, and action in the `fmdump` output, use the `-m` option. See the `fmdump`(1M) man page for more information.

EXAMPLE 2-4 Identifying Which CPUs Are Offline

Use the `psrinfo` command to display information about the CPUs:

```
$ psrinfo
0       faulted   since 05/13/2013 12:55:26
```

```
1       on-line   since 05/12/2013 11:47:26
```

The `faulted` state in this example indicates that the CPU has been taken offline by a Fault Manager response agent.

Displaying Information About Defective Services

The `fmadm faulty` command can also display information about problems in SMF services, as shown in the following example:

```
# fmadm faulty
--------------- ------------------------------------  -------------- ---------
TIME            EVENT-ID                              MSG-ID         SEVERITY
--------------- ------------------------------------  -------------- ---------
May 12 22:52:47 915cb64b-e16b-4f49-efe6-de81ff96fce7  SMF-8000-YX    major

Host        : parity
Platform    : Sun-Fire-V40z     Chassis_id  : XG051535088
Product_sn  : XG051535088

Fault class : defect.sunos.smf.svc.maintenance
Affects     : svc:///system/intrd:default
                  faulted and taken out of service
Problem in  : svc:///system/intrd:default
                  faulted and taken out of service

Description : A service failed - it is restarting too quickly.

Response    : The service has been placed into the maintenance state.

Impact      : svc:/system/intrd:default is unavailable.

Action      : Run 'svcs -xv svc:/system/intrd:default' to determine the
              generic reason why the service failed, the location of any
              logfiles, and a list of other services impacted. Please refer to
              the associated reference document at
              http://support.oracle.com/msg/SMF-8000-YX for the latest service procedures
              and policies regarding this diagnosis.
```

Follow the instructions given in the Action section in the `fmadm` output to display information about the defective service.

```
# svcs -xv svc:/system/intrd:default
svc:/system/intrd:default (interrupt balancer)
 State: maintenance since Wed May 12 22:52:47 2010
Reason: Restarting too quickly.
   See: http://support.oracle.com/msg/SMF-8000-YX
   See: man -M /usr/share/man -s 1M intrd
   See: /var/svc/log/system-intrd:default.log
Impact: This service is not running.
```

The references in the “See” lines provide more information about this problem.

♦ ♦ ♦ CHAPTER 3

Repairing Faults

This chapter discusses the following topics:

- How to repair faults or defects
- What information the various fault management log files contain and how to view those log files
- How to view information about Fault Manager modules

Repairing Faults or Defects

You can configure Oracle Auto Service Request (ASR) to automatically request Oracle service when specific hardware problems occur. See http://www.oracle.com/asr for more information about ASR.

When a component in your system has faulted, the Fault Manager can repair the component implicitly or you can repair the component explicitly.

Implicit repair

An *implicit repair* can occur when the faulty component is replaced or removed if the component has serial number information that the Fault Manager daemon (`fmd`) can track. On many SPARC based systems, serial number information is included in the FMRIs so that `fmd` can determine when components have been removed from operation, either through replacement or through other means such as blacklisting. When `fmd` determines that a component has been removed from operation, the Fault Manager no longer displays that component in `fmadm faulty` output. The component is maintained in the Fault Manager internal resource cache until the fault event is 30 days old.

When `fmd` detects faulty CPU or memory resources, those resources are placed on a blacklist. A faulty resource that is on the blacklist cannot be reassigned until `fmd` marks the resource as being repaired.

Explicit repair

Sometimes no FRU serial number information is available even though the FMRI includes a chassis identifier. In this case, `fmd` cannot detect an FRU replacement, and you must

perform an *explicit repair* by using the `fmadm` command with the `replaced`, `repaired`, or `acquit` subcommand as shown in the following sections. You should perform explicit repairs only at the direction of a specific documented repair procedure.

These `fmadm` commands take the following operands:

- The *UUID*, also shown as the `EVENT-ID` in Fault Manager output, identifies the fault event. The *UUID* can only be used with the `fmadm acquit` command. You can specify that the entire event can be safely ignored, or you can specify that a particular resource is not a suspect in this event.
- The *FMRI* and the *label* identify the suspect faulted resource. Examples of the FMRI and label of a resource are shown in Example 2-1. Typically, the label is easier to use than the FMRI.

A case is considered repaired when the fault event UUID is acquitted or when all suspect resources have been repaired, replaced, removed, or acquitted. A case that is repaired moves into the `FMD_CASE_REPAIRED` state, and the Fault Manager generates a `list.repaired` event.

`fmadm replaced` Command

Use the `fmadm replaced` command to indicate that the suspect FRU has been replaced or removed. If multiple faults are currently reported against one FRU, the FRU shows as replaced in all cases.

`fmadm replaced` *FMRI* `|` *label*

When an FRU is replaced, the serial number of the FRU changes. If `fmd` automatically detects that the serial number of an FRU has changed, the Fault Manager behaves in the same way as if you had entered the `fmadm replaced` command. If `fmd` cannot detect whether the serial number of the FRU has changed, then you must enter the `fmadm replaced` command if you have replaced or removed the FRU. If `fmd` detects that the serial number of the FRU has not changed, then the `fmadm replaced` command exits with an error.

If you remove the FRU but do not replace the FRU, the Fault Manager displays the suspect as `not present`. The suspect is not considered to be permanently removed until the fault event is 30 days old.

`fmadm repaired` Command

Use the `fmadm repaired` command when you have performed a physical repair other than replacement of the FRU to resolve the problem. Examples of such repairs include reseating a card or straightening a bent pin. If multiple faults are currently reported against one FRU, the FRU shows as repaired in all cases.

```
fmadm repaired FMRI | label
```

`fmadm acquit` Command

Use the `acquit` subcommand if you determine that the indicated resource is not the cause of the fault. Usually the Fault Manager automatically acquits some suspects in a multi-element suspect list. Acquittal can occur implicitly as the Fault Manager refines the diagnosis, for example if additional error events occur. Sometimes Support Services gives you instructions to perform a manual acquittal.

Replacement takes precedence over repair, and both replacement and repair take precedence over acquittal. Thus, you can acquit a component and then subsequently repair the component, but you cannot acquit a component that has already been repaired.

If you do not specify any *FMRI* or *label* with the *UUID*, then the entire event is identified as able to be ignored. A case is considered repaired when the fault event UUID is acquitted.

```
fmadm acquit UUID
```

Acquit by FMRI or label with no UUID only if you determine that the resource is not a factor in any current cases in which that resource is a suspect. If multiple faults are currently reported against one FRU, the FRU shows as acquitted in all cases.

```
fmadm acquit FMRI
fmadm acquit label
```

To acquit a resource in one case and keep that resource as a suspect in other cases, specify both the fault event UUID and the resource FMRI or both the UUID and the resource label, as shown in the following examples:

```
fmadm acquit FMRI UUID
fmadm acquit label UUID
```

Fault Management Log Files

The Fault Manager daemon records information in several log files. The log files are stored in `/var/fm/fmd`. To view these log files, use the `fmdump` command. See the `fmdump`(1M) man page for more information.

- The `errlog` log file records inbound telemetry information which consists of ereports.
- Informational events are recorded in two log files. The `infolog_hival` log file is for high-value events, and the `infolog` log file collects all other informational events.
- The `fltlog` log file records fault diagnosis and repair events.

Tip - Base your administrative action on output from the `fmadm faulty` command. Log files output by the `fmdump` command can contain error statements that are not faults or defects.

See "Displaying Information About Faults or Defects" on page 13 for information about using the `fmadm faulty` command.

The log files are automatically rotated. See the `logadm`(1M) man page for more information.

Fault Statistics

The Fault Manager daemon and many of its modules gather statistics. The `fmadm config` command shows the status of Fault Manager modules. The `fmstat` command reports statistics gathered by these modules.

```
# fmadm config
MODULE                   VERSION STATUS  DESCRIPTION
cpumem-retire            1.1     active  CPU/Memory Retire Agent
disk-diagnosis           0.1     active  Disk Diagnosis engine
disk-transport           2.1     active  Disk Transport Agent
eft                      1.16    active  eft diagnosis engine
ext-event-transport      0.2     active  External FM event transport
fabric-xlate             1.0     active  Fabric Ereport Translater
fmd-self-diagnosis       1.0     active  Fault Manager Self-Diagnosis
fru-monitor              1.1     active  FRU Monitor
io-retire                2.0     active  I/O Retire Agent
network-monitor          1.0     active  Network monitor
sensor-transport         1.2     active  Sensor Transport Agent
ses-log-transport        1.0     active  SES Log Transport Agent
software-diagnosis       0.1     active  Software Diagnosis engine
software-response        0.1     active  Software Response Agent
sysevent-transport       1.0     active  SysEvent Transport Agent
syslog-msgs              1.1     active  Syslog Messaging Agent
zfs-diagnosis            1.0     active  ZFS Diagnosis Engine
zfs-retire               1.0     active  ZFS Retire Agent
```

Without options, the `fmstat` command provides a high-level overview of the events, processing times, and memory usage of all loaded modules.

```
# fmstat
module             ev_recv ev_acpt wait  svc_t    %w  %b  open solve  memsz  bufsz
cpumem-retire            0       0  0.0 10010.0    0   0     0     0      0      0
disk-diagnosis           0       0  0.0 10007.7    0   0     0     0      0      0
disk-transport           0       0  0.9 1811945.5 92   0     0     0    52b      0
eft                      0       0  0.0 4278.0     0   0     3     0   1.6M    58b
ext-event-transport      6       0  0.0  860.8     0   0     0     0    46b   2.0K
fabric-xlate             0       0  0.0    4.8     0   0     0     0      0      0
fmd-self-diagnosis     393       0  0.0   25.5     0   0     0     0      0      0
fru-monitor              2       0  0.0   42.4     0   0     0     0   880b      0
```

```
io-retire               1       0  0.0 5003.8     0   0     0     0      0      0
network-monitor         0       0  0.0   13.2     0   0     0     0   664b      0
sensor-transport        0       0  0.0   38.3     0   0     0     0    40b      0
ses-log-transport       0       0  0.0   23.8     0   0     0     0    40b      0
software-diagnosis      0       0  0.0 10010.0    0   0     0     0   316b      0
software-response       0       0  0.0 10006.8    0   0     0     0    14K    14K
sysevent-transport      0       0  0.0 6125.0     0   0     0     0      0      0
syslog-msgs             2       0  0.0 3337.2     0   0     0     0      0      0
zfs-diagnosis           4       0  0.0 2002.0     0   0     0     0      0      0
zfs-retire              4       0  0.0 2715.1     0   0     0     0     4b      0
```

`ev_recv` The number of telemetry events received by the module.

`ev_acpt` The number of telemetry events accepted by the module as relevant to a diagnosis.

`wait` The average number of telemetry events waiting to be examined by the module.

`svc_t` The average service time for telemetry events received by the module, in milliseconds.

`%w` The percentage of time that telemetry events were waiting to be examined by the module.

`%b` The percentage of time that the module was busy processing telemetry events.

`open` The number of active cases (open problem investigations) owned by the module. The `open` column applies only to fault management cases, which are created and solved only by diagnosis engines. This column does not apply to other modules, such as response agents.

`solve` The total number of cases solved by this module since it was loaded. The `solve` column applies only to fault management cases, which are created and solved only by diagnosis engines. This column does not apply to other modules, such as response agents.

`memsz` The amount of dynamic memory currently allocated by this module.

`bufsz` The amount of persistent buffer space currently allocated by this module.

Different statistics and columns are displayed when you specify different options.

To display statistics on an individual module, use the `-m` *module* option. The `-z` option suppresses zero-valued statistics. The following example shows that the `cpumem-retire` response agent successfully processed a request to take a CPU offline.

```
# fmstat -z -m cpumem-retire
```

```
NAME      VALUE        DESCRIPTION
cpu_flts  1            cpu faults resolved
```

See the fmstat(1M) man page for information about other options.

Index

N

O

P

R

S