Oracle Hardware Management Pack 2.4 Linux Fault Management Architecture Software User's Guide

E72771-05
May 2023

Oracle Hardware Management Pack 2.4 Linux Fault Management Architecture Software User's Guide,

E72771-05

Primary Author: Ralph Woodley

Contents

3 Troubleshooting Oracle Linux Fault Management Architecture

Index

Using This Documentation

- **Overview** – Describes how to install the software
- **Audience** – Technicians, system administrators, and authorized service providers
- **Required knowledge** – Advanced experience troubleshooting and replacing hardware
- Product Documentation Library
- Feedback
- Change History

Product Documentation Library

Documentation and resources for this product and related products are available at https://www.oracle.com/goto/ohmp/docs.

Feedback

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

Change History

The following changes have been made to the documentation set.

- April 2017. Initial publication.
- July 2017. Updated for Oracle Hardware Management Pack 2.4.1 to add new `fmadm list` and `fmadm clear` subcommands.
- May 2021. Updated to specify that Oracle Linux FMA command line tools interact directly with physical hardware and can not be installed or run in a virtual environment.
- May 2023. Minor document formatting edits.

1

Installing the Oracle Linux Fault Management Architecture Software

The Oracle Linux Fault Management Architecture (FMA) software can be used to manage server faults from the host OS in much the same way you manage faults in the Oracle Integrated Lights Out Manager (ILOM). In addition, CPU or memory faults detected by the Oracle Linux OS mcelog daemon are diagnosed, converted to a standard fault format, and stored in a fault management database on the host that is synced with the fault management database in Oracle ILOM. This provides detailed fault information an administrator or service personnel can act on from the host OS utilizing a set of fault management commands similar to those found in Oracle ILOM.

This section describes the prerequisites and installation of Oracle Linux FMA software:

Requirements

This section lists the requirements for installing Oracle Linux FMA.

- Check to see if Oracle Linux FMA is supported for use on your server. See the support matrix available at:

  http://www.oracle.com/goto/ohmp

- Oracle Linux 6.5 or later must be installed on the server. In addition,
  - The IPMI service must be running.
  - The dmidecode package must be installed.
  - The mcelog package must be installed and running.
- Oracle Hardware Management Pack 2.4 or later must be installed with the following components selected or configured:
  - Oracle Linux FMA software
  - Configure Host-to-ILOM Interconnect (automatic is preferred)
  - CLI Tools (specifically, `ilomconfig` if you do not configure Host-to-ILOM interconnect using the Oracle Hardware Management Pack installer)
- Oracle Linux FMA commands interact directly with physical hardware and can not be installed or run in a virtual environment.

For more information, see How to Install the Oracle Linux FMA Software.

How to Install the Oracle Linux FMA Software

This section describes the installation process for Oracle Linux FMA software. Actual instructions are provided in the .

The installation process includes the following steps.

1. Install the Required Linux Components Before Installing Oracle Linux FMA Software in *Oracle Hardware Management Pack 2.4 Installation Guide*
2. Choose an installation method:
   - Installing Components Using the Oracle Hardware Management Pack Installer in *Oracle Hardware Management Pack 2.4 Installation Guide*

     --or--

   - Installing and Uninstalling Components Manually on a Linux Server in *Oracle Hardware Management Pack 2.4 Installation Guide*
3. Confirm That the Oracle Linux FMA Software is Running in *Oracle Hardware Management Pack 2.4 Installation Guide*

2

Using the Oracle Linux Fault Management Architecture Software

The Oracle Linux OS includes an architecture for building and deploying systems and services that are capable of predictive self-healing.

The Oracle Linux Fault Management Architecture (FMA) software is an add-on service that receives data related to hardware errors detected by the host OS (CPU and memory) and automatically diagnoses the underlying problem. These fault diagnosis messages are maintained in a fault management database at the host that is synced with the Oracle ILOM fault management database to allow access to all system faults from either the host OS or Oracle ILOM. Fault management commands, similar to those available with Oracle ILOM, allow access to the database for viewing and clearing of diagnosed faults from the host OS.

The following topics are covered in this section:

Fault Management Architecture Overview

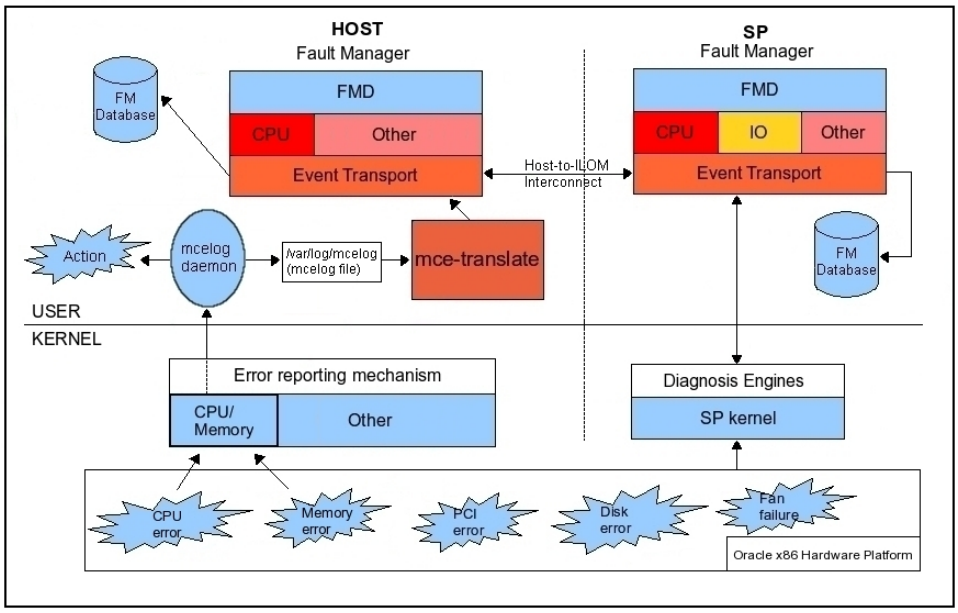
The Oracle Linux Fault Management Architecture (FMA) is a supplement to existing Linux error detecting and recovery mechanisms that allows system administrators to access view, act on, and clear faults detected by the Linux Kernel from the host. It uses the Oracle ILOM fault manager on the service processor to diagnose CPU and memory errors captured from the host and output them to a standard fault format that is stored in a fault management database.

This database contains a superset of all detected faults (those captured by Oracle Linux FMA and Oracle ILOM FMA) and is maintained on both the host and service processor.

In the Oracle Linux operating system, CPU and memory errors are generated at the kernel level as machine check events. These events are stored in the Linux mcelog database. The Linux mcelog daemon, `mcelogd`, retrieves errors stored in the database and converts them to human-readable messages that are output to the console, the mcelog file (`/var/log/mcelog`), and to the Linux system log. The mcelog daemon also takes action based on a set of rules stored in a configuration file. For example, these actions might include retiring a memory page from service because it contains uncorrectable errors.

The information logged by mcelog might not contain enough information to identify a bad component (such as a memory DIMM). The Oracle Linux FMA Fault Manager daemon, `fmd`, scans and retrieves errors stored in the mcelog file and translates the errors into the ereport format supported by Oracle ILOM. It then forwards the ereport to the service processor using the internal Host-to-ILOM interconnect port. The Oracle ILOM fault manager uses the ereport to diagnose the fault. Oracle ILOM then logs the fault in its own fault management database and sends a copy to the fault management database that resides on the Linux host.

Using this method, all system hardware faults in the database can be viewed and acted on using a similar set of fault management commands whether from the host OS or from Oracle ILOM.

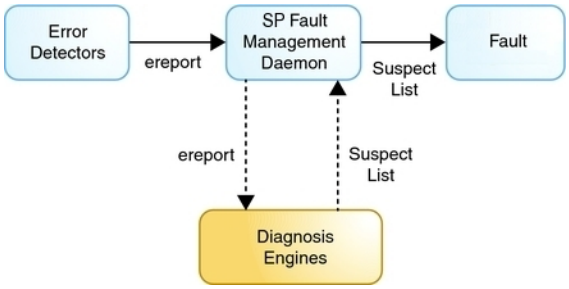


Oracle server platforms running Oracle Linux include error detectors, diagnosis engines, and response agents. Error detectors and response agents reside on the Oracle Linux host. The diagnosis engines reside on the server's service processor.

- **Error detectors** – These detect errors in the system and perform any immediate, required handling. They also generate well-defined error reports, or *ereports*, to a diagnosis engine. In Linux, the mcelog daemon detects errors, and the Oracle Linux Fault Management software collects and reformats them into ereports and forwards them to the service processor for fault diagnosis.
- **Diagnosis engines** – A set of diagnostic engines located on the service processor interpret reports and determine whether a fault or defect is present. When such a determination is made, the diagnosis engine creates a *suspect list* that describes the resource or set of resources that might be the cause of the problem. The resource might or might not have an associated field-replaceable unit (FRU) or a label.

  When the suspect list includes multiple suspects, for example, if the diagnosis engine cannot isolate a single suspect, the suspects are assigned a probability as to each suspect being the key suspect. The probabilities in this list add up to 100 percent.

Error detectors and diagnosis engines are connected by the Fault Manager daemon on the service processor, which acts as a multiplexor between the various components, as shown in the following figure.



- **Response agents** – These agents attempt to take action based on the type of error. On the host side, the mcelog daemon acts as the response agent. Responses include logging messages and retiring memory pages.

The Oracle Linux Fault Manager daemon, fmd, is itself a service. The service can be enabled and controlled as a scriptless daemon, or by using `init.d` scripts for greater manageability. Fault management commands supported in this version of Oracle Linux FMA include:

- `fmadm` – Used by administrators and service personnel to view and clear faults maintained by the Oracle Linux Fault Manager, `fmd`.
- `fmdump` – Used to display the contents of any of the log files associated with the Oracle Linux Fault Manager, `fmd`.

Lifecycle of a Problem or Condition Managed By the Fault Manager

The lifecycle of a problem or condition managed by the Fault Manager can include the following stages. Each of these lifecycle state changes is associated with the publication of a unique list event.

- **Diagnose** – A new diagnosis has been made by the Fault Manager. The diagnosis includes a list of one or more suspects. A `list.suspect` event is published. The diagnosis is identified by a UUID in the event payload, and further events describing the resolution lifecycle of this diagnosis quote a matching UUID.
- **Isolate** – A suspect has been automatically isolated to prevent further errors from occurring. A `list.isolated` event is published. For example, a CPU core or memory page has been offlined.
- **Update** – One or more of the suspect resources in a problem diagnosis has been repaired, replaced, or acquitted, or the resource has faulted again. A `list.updated` event is published. 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 Repairing Faults and Defects and Clearing Alerts.
- **Repair** – All of the suspect resources in a diagnosis have been repaired, resolved, or acquitted. A `list.repaired` event is published. Some or all of the resources might still be isolated.

- **Resolve** – All of the suspect resources in a diagnosis have been repaired, resolved, or acquitted and are no longer isolated. A `list.resolved` event is published. For example, a CPU core 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 enabled by default when using the Oracle Hardware Management Pack installer. See the `fmd` 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, isolate resources, generate notifications, and auto-repair problems in the system.

Fault Management Architecture Terms

The following table contains descriptions for Fault Management Architecture terms used in this document.

Term	Description
CRU	A CRU is a customer-replaceable unit (such as a memory DIMM).
Diagnosis class	The diagnosis class is a unique identifier of the form sub-class1.sub-class2...sub-classN that uniquely identifies the type of fault, defect, or alert event associated with a diagnosis. The diagnosis class is also called the problem class.
Diagnosis engines	Oracle Linux FMA utilizes diagnosis engines that reside on the service processor to process hardware event ereports, including those generated by mcelog. For a list of diagnosis engines supported in the fault management architecture for Oracle ILOM, see the Oracle ILOM documentation.
Error report (Ereport)	Error reports describe error events. They include raw device and error type information so that the fault manager can diagnose the error and create an appropriate fault diagnosis message.
Fault	A fault indicates that a hardware component is present but is unusable or degraded because one or more problems have been diagnosed by the fault manager. The component has been disabled to prevent further damage to the system.
Fault case	When problems are diagnosed, the fault manager logs a fault diagnosis message that contains a case id (represented by a UUID) which references the problem.
FRU	A FRU is a field-replaceable unit (such as a processor).
Label	A location string (also called a FRU label), such as "/SYS/MB/P1" which represents the processor #1 located on the motherboard of the system. The quoted value is intended to match the label on the physical hardware or when viewed in Oracle ILOM.

Term	Description
Machine check events	Platform error(s) detected by the hardware and reported to the OS. The error reported might be correctable or uncorrectable, recoverable or fatal. In Linux, the mcelog captures these errors.
mcelog	mcelog provides error handling and predictive failure analysis in x86 Linux systems. The mcelog daemon processes CPU and memory machine check events and executes actions based on configurable error thresholds. A range of actions can be configured, including bad memory page retirement, CPU core offlining, and automatic cache error handling. User defined actions can be also configured. Oracle Linux FMA captures errors processed by mcelog and stored in the mcelog log file, converts them to a standard Oracle fault format, and adds them to a synced fault management database available on both the host and Oracle ILOM.
Page retirement	A kernel facility in newer Linux OSes where an OS memory page corresponding to a defective physical memory location is removed from service, if possible. This feature helps increase system availability.
Proactive self-healing	Proactive self-healing is a fault management architecture and methodology for automatically diagnosing, reporting, and handling software and hardware fault conditions. Proactive self-healing reduces the time required to debug a hardware or software problem and provides the system administrator or Oracle Services personnel with detailed data about each fault. The architecture consists of the Linux mcelog event management protocol, the Fault Manager, and service processor-based diagnosis engines that process errors received from the host OS to a standard FMA fault case.
Resource	A resource is a physical or abstract entity in the system against which diagnoses can be made.
Service processor (SP)	Most Oracle servers ship with a service processor that controls chassis functions such as power budgeting and control, system health monitoring, and FMA activities including error reporting and fault diagnosis.
Universal unique identifier (UUID)	A UUID is used to uniquely identify a problem across any set of systems.

Notification of Faults, Defects and Alerts

When the mcelog daemon encounters an error, it triggers a configurable response and logs information to the mcelog file. For example, assume that physical address location 0x45a3b50c0 generates a correctable memory read error. When this happens, the mcelog daemon adds an entry to `/var/log/mcelog` . For example:

```
CPU 8
BANK 3
TSC 0
RIP 00:0
```

```
MISC 0x85
ADDR 0x45a3b50c0    <------ address that had the correctable read error
STATUS 0x9c000000f00c009f
MCGSTATUS 0x7
PROCESSOR 0:0x306f1
TIME 1389814624
SOCKETID 0
APICID 18
MCGCAP 0x7000c16
```

A message is also sent to the system log (`/var/log/messages`) describing the problem (error count exceeded threshold) and what was done (offlining the page), such as:

```
1  Jan 15 14:37:04 testserver16 kernel: Machine check poll done on CPU 8
2  Jan 15 14:37:04 testserver16 mcelog: Family 6 Model 3f CPU: only decoding
architectural errors
3  Jan 15 14:37:04 testserver16 mcelog: corrected Socket memory error count
exceeded threshold: 1 in 24h
4  Jan 15 14:37:04 testserver16 mcelog: Location SOCKET:0 CHANNEL:? DIMM:? []
5  Jan 15 14:37:04 testserver16 mcelog: Corrected memory errors on page
45a3b5000
exceed threshold 1 in 24h: 1 in 24h
6  Jan 15 14:37:04 testserver16 mcelog: Location SOCKET:0 CHANNEL:? DIMM:? []
7  Jan 15 14:37:04 testserver16 mcelog: Running trigger `page-error-trigger'
8  Jan 15 14:37:04 testserver16 mcelog: Offlining page 45a3b5000
```

The message on line 5 indicates that the correctable error threshold was set to 1 error in 24 hours. Since this threshold was exceed, the action taken was to remove page 0x45a3b5000 from service. This is indicated by the "Offlining page" message (line 8) in the system log. The process that encountered the correctable error is either assigned a new page, or it is killed, depending on the "memory-ce-action" value in the "page" section of the `mcelog.conf` file.

In addition to the page being offlined, if the DIMM corresponding to the failed address exceeds the factory programmed DIMM threshold, the SP generates a fault that is forwarded to the host and logged as part of the fault management database.

Often, the first interaction with the Fault Manager daemon is a system message indicating that a fault or defect has been diagnosed. Messages are sent to both the console and the `/var/log/messages` file. All messages from the Fault Manager daemon use the following format:

```
1    SUNW-MSG-ID: SPX86A-8002-30, TYPE: Fault, VER: 1, SEVERITY: Minor
2    EVENT-TIME: Wed Nov 27 10:36:30 PST 2013
3    PLATFORM: SUN SERVER X4-4, CSN: -, HOSTNAME: testserver16
4    SOURCE: fdd, REV: 1.0
5    EVENT-ID: eed2208e-2dcf-40c9-9bab-ab3a13e94182
6    DESC: A processor has detected multiple memory controller correctable
     errors.
8    AUTO-RESPONSE: The affected processor will be disabled at the next system
boot
9    and remain unavailable until repaired.
10   The chassis wide and processor service-required LED's are illuminated.
11   IMPACT: The system will continue to operate in the presence of this
12   fault.
13   System performance may be impacted due to disabled processor.
14   REC-ACTION: Use 'fmadm faulty' to provide a more detailed view of this
15   event. Please refer to the associated reference document at
16   http://support.oracle.com/msg/SUN4V-8001-8H for the latest service
```

```
procedures and
17   policies regarding this diagnosis.
```

When notified of a diagnosed problem, always consult the recommended Oracle Knowledge Article for additional details. See line 16 above for an example. The knowledge article might contain additional actions that you or a service provider should take beyond those listed on line 14.

Notification of events can also be configured in Oracle ILOM using the Simple Network Management Protocol (SNMP) or the Simple Mail Transfer Protocol (SMTP). See the Oracle ILOM documentation at: http://www.oracle.com/goto/ilom/docs

In addition, Oracle Auto Service Request can be configured to automatically request Oracle service when specific hardware problems occur from supported telemetry resources (such as Oracle ILOM). See the Oracle Auto Service Request product page for information about this feature. The documentation link on this page provides links to *Oracle ASR Quick Installation Guide* and *Oracle ASR Installation and Operations Guide*.

Paths to Oracle Linux FMA Commands and Man Pages

When Oracle Linux FMA is installed, no path variables are set in the operating system for commands or man pages. This means that Oracle Linux FMA commands and man pages must be run with the full path.

For example, to run the `fmadm` command, enter:

```
/opt/fma/fm/sbin/fmadm
```

To launch a man page, enter:

```
man -S 1m -M :/opt/fma/share/man: fmadm
```

For ease of use, you can add these to your PATH and MANPATH environmental variables as described in the following table.

Item	Installation Path	Setting

Commands and binaries	`/opt/fma/fm/sbin`	• For the Bourne Shell (sh), Bourne-again Shell (bash), or Korn shell (ksh), enter:
		`$ PATH=$PATH:/opt/fma/fm/sbin; export PATH`
		• For the C Shell (csh) or enhanced C Shell (tcsh), enter:
		`% setenv PATH $PATH:/opt/fma/fm/sbin`

Item	Installation Path	Setting

Man pages	`/opt/fma/share/man`	• For the Bourne Shell (sh), Bourne-again Shell (bash), or Korn shell (ksh), enter:
		`$ MANPATH=$MANPATH:/opt/fma/share/man; export MANPATH`
		• For the C Shell (csh) or enhanced C Shell (tcsh), enter:
		`% setenv MANPATH $MANPATH:/opt/fma/share/man`

Displaying Information About Faults, Defects and Alerts

The following commands display detailed information about diagnoses made by the fault management system.

- The `fmadm list` command displays all active faults, defects, and alerts.

> **Note:**
>
> The `fmadm faulty` command has been deprecated and is replaced by `fmadm list`.

- The `fmadm list-fault` command displays all active faults.
- The `fmadm list-defect` command displays all active defects.
- The `fmadm list-alert` command displays all active alerts.

The `fmadm list` commands are the preferred method to display fault, defect or alert information and determine the FRUs involved. However, the `fmdump` command is also supported. Normally, `fmadm list` is used to display active problems and `fmdump` is used to display an historical log of problems on the system.

> **Caution:**
>
> Do not base administrative action on the output of the `fmdump` command. Instead, use the `fmadm list` output. The log files can contain error statements, which should not be considered faults or defects.

Display Information About Faulty Components

1. Become an administrator.
2. To display information about the components, enter the command:

   ```
   fmadm list-fault
   ```

   See the following example for a description of the text generated.

Example 2-1 `fmadm` Output With One Faulty CPU

```
# fmadm list-fault
------------ ------------------------------------  -------------- ---------
TIME         EVENT-ID                              MSG-ID         SEVERITY
------------ ------------------------------------  -------------- ---------
Dec 19 13:25 eed2208e-2dcf-40c9-9bab-ab3a13e94182  SPX86A-8002-30 Minor

Problem Status    : open
Diag Engine       : fdd 1.0
System
   Manufacturer   : Oracle Corporation
   Name           : SUN SERVER X4-4
   Part_Number    : 7066596
   Serial_Number  : 489089M+13280X0042
   Host_ID        : testserver16

 ----------------------------------------
 Suspect 1 of 1
    Fault class  : fault.cpu.intel.mc_ce
    Certainty    : 100%
    Affects      : /chassis=0/motherboard=0/chip=0
    Status       : faulted but still in service

    FRU
       Location          : "/SYS/MB/P0"
       Manufacturer      : Intel
       Name              : unknown
       Part_Number       : CM80636
       Revision          : unknown
       Serial_Number     : unknown
       Chassis
          Manufacturer   : Oracle Corporation
          Name           : SUN SERVER X4-4
          Part_Number    : 7066596
          Serial_Number  : 489089M+13280X0042
          Status         : faulty

 Description : A processor has detected multiple memory controller correctable
               errors.

 Response    : The affected processor will be disabled at the next system boot
               and remain unavailable until repaired.
               The chassis wide and processor service-required LED's are
               illuminated.

 Impact      : The system will continue to operate in the presence of this
               fault.
               System performance may be impacted due to disabled processor.

 Action      : Please refer to the associated reference document at
```

```
45                http://support.oracle.com/msg/SPX86A-8002-30 for the latest
46                service procedures and policies regarding this diagnosis.
```

The `Affects` and `Status` lines (lines 20 and 21) identify the component affected by the fault and its relative state. In this example, a single CPU is affected. It is marked "faulted but still in service".

Line 24 shows the data for the impacted FRU (Field Replaceable Unit). The location string (also called the FRU label) is "/SYS/MB/P0." it should match the label on the physical hardware or the value displayed by Oracle ILOM.

Line 35 shows the state as `faulty`.

The Action section might include other specific actions and/or references to additional information.

Example 2-2 Showing Faults with the `fmdump` Command

Some console messages and knowledge articles might instruct you to use the older `fmdump -v -u` *`uuid`* command to display fault information. Although the `fmadm list` command is preferred, the `fmdump` command still operates, as shown in the following example:

```
1   # fmdump -v -u eed2208e-2dcf-40c9-9bab-ab3a13e94182
2   TIME                 UUID                                 SUNW-MSG-ID
3   Dec 19 13:25:38.0697 eed2208e-2dcf-40c9-9bab-ab3a13e94182 SPX86A-8002-30
4     100%  fault.cpu.intel.mc_ce
5
6     Problem in: hc://:chassis-serial=489089M+13280X0042:chassis-name=SUN
SERVER X4-4:
chassis-part=7066596:chassis-mfg=Oracle Corporation/chassis=0/motherboard=0/
chip=0
7        Affects: hc://:chassis-serial=489089M+13280X0042:chassis-name=SUN
SERVER X4-4:
chassis-part=7066596:chassis-mfg=Oracle Corporation/chassis=0/motherboard=0/
chip=0
8            FRU: hc://:chassis-serial=489089M+13280X0042:chassis-name=SUN
SERVER X4-4:
chassis-part=7066596:chassis-mfg=Oracle Corporation:fru-serial=:fru-part=CM80636:
fru-revision=/chassis=0/motherboard=0/chip=0
9       Location: /SYS/MB/P0
```

The time the fault was generated, its Unique Universal Identifier (UUID) and message ID are displayed on line 3.

The fault and percent certainty of the diagnosis are displayed on line 4.

If available, the FRU is presented in a Fault Management Resource Identifier (FMRI) format (lines 6 through 8), which includes descriptive properties about the system containing the fault, such as its chassis name (`SUN SERVER X4-4`) and chassis serial number. On platforms and components that support it, the part number and serial number of the FRU are also included in the FRU's FMRI. Otherwise, the FRU label is shown.

The FRU location (line 9) presents the human-readable FRU label. For example, the label `/SYS/MB/P0` represents the CPU labeled "P0" on the motherboard.

Note that the severity, descriptive text, and action are not shown with the `fmdump` command, unless you use the `-m` option. See the `fmdump` man page for more information.

Display Information About Defective Services

The `fmadm list-defect` command can display information about problems in SMF services.

1. Become an administrator.
2. To display information about the components, enter the command:

   ```
   fmadm list-defect
   ```

   See the following example for a description of the text generated.

Example 2-3 `fmadm list-defect` Output

The following example shows a defect with a system's Memory Reference Code (MRC):

```
# fmadm list-defect
-------------- ------------------------------------  -------------- ---------
TIME           EVENT-ID                              MSG-ID         SEVERITY
-------------- ------------------------------------  -------------- ---------
Apr 29 2000    14cfc32b-fc99-47c3-975c-daac16863ff5  SPX86A-8005-L6 Critical

Problem Status    : open
Diag Engine       : fdd / 1.0
System
    Manufacturer  : Oracle Corporation
    Name          : ORACLE SERVER X6-2L
    Part_Number   : 1234567
    Serial_Number : 1234567
    Host_ID       : (null)

----------------------------------------
Suspect 1 of 1 :
   Problem class : defect.memory.intel.mrc.internal
   Certainty   : 100%

   FRU
     Status           : faulty
     Location         : "/SYS"
     Manufacturer     : Oracle Corporation
     Name             : ORACLE SERVER X6-2L
     Part_Number      : 1234567
     Revision         : unknown
     Serial_Number    : X6-2L_014
     Chassis
        Manufacturer  : Oracle Corporation
        Name          : ORACLE SERVER X6-2L
        Part_Number   : 1234567
        Serial_Number : X6-2L_014
   Resource
     Status           : faulted but still in service

Description : An error in the Memory Reference Code (MRC) has been detected.

Response    : The chassis wide service-required LED will be illuminated.

Impact      : The system is unable to power on.
```

```
Action      : Please refer to the associated reference document at
              http://support.oracle.com/msg/SPX86A-8005-L6 for the latest
              service procedures and policies regarding this diagnosis.
```

Display Information About Alerts

An *alert* is information of interest that is neither a fault nor a defect. An alert might report a problem or might be simply informational. A problem that is reported by an alert is a misconfiguration or other problem that the administrator can resolve without assistance from a response agent. An example of this type of problem is a DIMM plugged into the wrong slot. An example of an informational message reported by an alert is a message that a shadow migration has completed.

The following list provides examples of alert messages:

- **Threshold alerts** – Temperature is high, storage is at capacity, a quota is exceeded, the path count to a chassis or disk has changed. These kinds of alerts can predict a performance impact.
- **Configuration checks** – A FRU has been added or removed, SAS cabling is incorrect, a DIMM is plugged into the wrong slot, a datalink changed, a link went up or down, Oracle ILOM is misconfigured, MTU (Maximum Transmission Unit - TCP/IP) is misconfigured.
- **Interesting events** – A reboot occurred, file system events occurred, firmware has been upgraded, save core failed.

Alerts can be in one of the following states:

- `active` – The alert has not been cleared.
- `cleared` – The alert has been cleared. The `cleared` state for alerts can be compared to the `resolved` state for faults and defects. See the following description of persistent and transient alerts for more information about clearing an alert.

Alerts can be persistent or transient.

- A persistent alert is active until it is manually cleared as shown in fmadm clear Command.
- A transient alert clears after a specified timeout period or is cleared by a service such as a network monitor.

1. Become an administrator.
2. To display information about the components, enter the command:

   ```
   fmadm list-alert
   ```

> **Tip:**
>
> Base your administrative action on output from the `fmadm list-alert` command. Log files output by the `fmdump` command contain a historical record of events and do not necessarily present active or open diagnoses. Log files output by `fmdump -i` are a historical record of telemetry and might not have been diagnosed into alerts.

See the following example for a description of the text generated.

Example 2-4 `fmadm list-alert` Output

Use the `fmadm list-alert` command to list all alerts that have not been cleared. The following alert shows that Top Level Identifier information (TLI) for the system is missing or corrupted. The Problem Status has the value `open`, which is an active state. Problem Status can be `open`, `isolated`, `repaired`, or `resolved`. The Problem class indicates that the chassis TLI is invalid. The Impact indicates how the system might be impacted by the issue. Perhaps the most useful piece of information in this output is the MSG-ID. Follow the instructions in the Action at the end of the alert to access more information about `SPX86A-8006-5T`.

```
# fmadm list-alert
------------------ ------------------------------------ -------------- --------
Time               UUID                                 msgid          Severity
------------------ ------------------------------------ -------------- --------
2000-06-01/08:40:14 5f252c60-0668-e32a-f0de-b3e9f24228df SPX86A-8006-5T Critical

Problem Status           : open [injected]
Diag Engine              : fdd 1.0
System
   Manufacturer          : Oracle Corporation
   Name                  : ORACLE SERVER X7-2L
   Part_Number           : 1234567
   Serial_Number         : 1234567

System Component
   Firmware_Manufacturer : Oracle Corporation
   Firmware_Version      : (ILOM)4.0.0.0
   Firmware_Release      : (ILOM)2017.06.02

----------------------------------------
Suspect 1 of 1
   Problem class  : alert.memory.intel.dimm.mismatch
   Certainty      : 100%
   Affects        : /SYS
   Status         : faulted

   FRU
      Status            : Active
      Location          : /SYS
      Manufacturer      : Oracle Corporation
      Name              : ORACLE SERVER X7-2L
      Part_Number       : 1234567
      Serial_Number     : X7-2L_014
      Chassis
         Manufacturer   : Oracle Corporation
         Name           : ORACLE SERVER X7-2L
         Part_Number    : 1234567
         Serial_Number  : X7-2L_014

Description : DIMMs of different types were detected.

Response    : The chassis wide service-required LED will be illuminated.

Impact      : The system is unable to power on.

Action      : Please refer to the associated reference document at
              http://support.oracle.com/msg/SPX86A-8006-5T for the latest
              service procedures and policies regarding this diagnosis.
```

Repairing Faults and Defects and Clearing Alerts

This section includes the following topics:

Repairing Faults and Defects

After Fault Management has identified a faulted component in your system, you should repair it. A repair can happen in one of two ways: implicitly or explicitly.

- **Implicit repair** – An *implicit repair* can occur when the faulty component is replaced or removed, provided the component has serial number information that the Fault Manager daemon can track. The system's serial number information is included so that the Fault Manager daemon can determine when components have been removed from operation, either through replacement or other means. When such detections occur, the Fault Manager daemon no longer displays the affected resource in `fmadm list` output. The resource is maintained in the daemon's internal resource cache until the fault event is 30 days old, at which point it is purged.
- **Explicit repair** – An *explicit repair* is required if no FRU serial number is available. For example, CPUs have no serial numbers. In these cases, the Fault Manager daemon cannot detect a FRU replacement.

  Use the `fmadm` command to explicitly mark a fault as repaired. The options include:

  – `fmadm replaced` *label*
  – `fmadm repaired` *label*
  – `fmadm acquit` *label* `[`*uuid*`]`
  – `fmadm acquit` *uuid*

  Although these four commands can take UUIDs or labels as arguments, it is better to use the label. For example, the label `/SYS/MB/P0` represents the CPU labeled "P0" on the motherboard.

  If a FRU has multiple faults against it and you want to replace the FRU only one time, use the `fmadm replaced` command against the FRU.

fmadm replaced Command

You can use the `fmadm replaced` command to indicate that the suspect FRU has been replaced or removed.

If the system automatically discovers that a FRU has been replaced (the serial number has changed), then this discovery is treated in the same way as if `fmadm replaced` had been typed on the command line. The `fmadm replaced` command is not

allowed if `fmd` can automatically confirm that the FRU has not been replaced (the serial number has not changed).

If the system automatically discovers that a FRU has been removed but not replaced, then the current behavior is unchanged: The suspect is displayed as `not present`, but is not considered to be permanently removed until the fault event is 30 days old, at which point it is purged.

fmadm repaired Command

You can use the `fmadm repaired` command when some physical repair has been carried out to resolve the problem, other than replacing a FRU. Examples of such repairs include reseating a component or straightening a bent pin.

fmadm acquit Command

Often you use the `acquit` option when you determine that the resource was not the cause. Acquittal can also happen implicitly when additional error events occur, and the diagnosis gets refined.

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

A case is considered repaired (moves into the `FMD_CASE_REPAIRED` state and a `list.repaired` event is generated) when either its UUID is acquitted, or all suspects have been either repaired, replaced, removed, or acquitted.

Usually `fmd` automatically acquits a suspect in a multi-element suspect list, or Oracle Support Services gives you instructions to perform a manual acquittal. You would only want to acquit by label if you determined that the resource was not guilty in all current cases in which it is a suspect. However, you can allow a FRU to be manually acquitted in one case while remaining a suspect in all others, using the following option which enables you to specify both UUID and label:

`fmadm acquit` *`uuid`* `[`*`label`*`]`

Clearing Alerts

Use the `fmadm list-alert` command to list all alerts that have not been cleared. See Display Information About Alerts for example output from the `fmadm list-alert` command.

Similar to faults, alerts can be repaired implicitly or explicitly. Because alerts do not necessarily represent problems that must be fixed, alerts are said to be cleared rather than repaired. An alert that is cleared is no longer active and no longer displayed by the `fmadm list` or `fmadm list-alert` commands.

- **Implicit clear** – An *implicit clear* occurs when the alert clears with no administrative action. For example, an alert that an FRU has been removed is automatically cleared by an alert that the same FRU has been added, and an alert that an FRU has been added automatically clears after 30 seconds.
- **Explicit clear** – Use the `fmadm clear` command to notify the Fault Manager that the specified alert event should be cleared.

fmadm clear Command

The `fmadm clear` command requires one of the following arguments:

```
fmadm clear uuid | location | class@resource
```

For the following examples, refer to the output from the `fmadm list-alert` command in Display Information About Alerts.

In the following example, *uuid* is the value of the EVENT-ID field at the top of the `fmadm list-alert` output:

```
# fmadm clear 5cd3380f-34db-ec51-c19f-e2eb6f612d3e
```

In the following example, *location* is the value of the FRU Location field in the `fmadm list-alert` output. This location is also referred to as the label.

```
# fmadm clear /SYS
fmadm: cleared alert /SYS
```

In the following example, *class* is the value of the Problem class field of the suspect in the `fmadm list-alert` output, and *resource* is the value of the `Problem in` field found using the `fmdump -vu` *uuid* command as shown in Example 2-2. Note that the command line in this example is artificially divided to improve readability.

```
# fmadm clear alert.ilom.chassis.tli.invalid@
                      hc://:chassis-serial=X6-2L_014:chassis-name=ORACLE
SERVER X6-2L
                      :chassis-part=1234567:chassis-mfg=Oracle
Corporation/chassis=0"
fmadm: cleared alert alert.ilom.chassis.tli.invalid@
hc://:chassis-serial=X6-2L_014:chassis-name=ORACLE SERVER X6-2L
:chassis-part=1234567:chassis-mfg=Oracle Corporation/chassis=0
```

Log Files and Fault Manager Status

This section includes the following topics:

Fault Management Log Files

The Fault Manager daemon, `fmd`, records information in several log files. The log files are stored in `/var/opt/fma/fm/fmd` and are viewed by using the `fmdump` command. See the `fmdump` 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. `infolog_hival` is for high-value events, and `infolog` collects all other informational events.

- The `fltlog` log file records fault diagnosis and repair events.

> **Caution:**
>
> Do not base administrative action on the contents of the log files, but rather on the `fmadm list` output. The log files can contain error statements, which should not be considered faults or defects.

The log files are automatically rotated. See the `logrotate(8)` man page for more information on managing log files in Oracle Linux.

Fault Manager Module Status

You can obtain the current status of the Fault Manager daemon. The `fmadm config` command shows the status of Fault Manager modules.

Example 2-5 `fmadm config` Output

```
# fmadm config

MODULE                   VERSION STATUS  DESCRIPTION
ext-event-transport      0.2     active  External FM event transport
fmd-self-diagnosis       1.0     active  Fault Manager Self-Diagnosis
ip-transport             1.1     active  IP Transport Agent
mce                      1.0     active  Machine Check Translator
sysevent-transport       1.0     active  SysEvent Transport Agent
syslog-msgs              1.1     active  Syslog Messaging Agent
```

3

Troubleshooting Oracle Linux Fault Management Architecture

This section describes problems and provides troubleshooting procedures for Oracle Linux FMA. It includes:

- Check Services and Modules
- Restart fmd if mcelog Fails
- Edit mcelog File if Faults Are Not Present in the Fault Management Database
- fmd Daemon Might Not Start if SELinux is Running
- Oracle Linux FMA Installation Can Fail When Using Either Anaconda or Oracle System Assistant to Install the OS

Check Services and Modules

- If Oracle Linux Fault Management (FMA) software is not working correctly, check that the following modules and services are in the correct state:

Oracle Linux Version	Service or Module	Required State
Oracle Linux 6.5 or later and 7.0 or later	IPMI service	Installed and running
Oracle Linux 6.5 or later and 7.0 or later	dmidecode	Installed and available
Oracle Linux 6.5 or later and 7.0 or later	EDAC module	Disabled
Oracle Linux 6.5 or later	mcelog service	Installed and running
Oracle Linux 7.0 or later	mcelog service	Installed and running in daemon mode only

For detailed information to check and configure the services and modules, refer to: Install the Required Linux Components Before Installing Oracle Linux FMA Software in *Oracle Hardware Management Pack 2.4 Installation Guide*.

Restart fmd if mcelog Fails

For various reasons, it is possible that the mcelog daemon might not start or fail during normal operation. When this happens, you stop receiving and diagnosing CPU and memory errors from the host.

1. Determine if the mcelog daemon is running.

   For example for Oracle Linux 6.5:

```
[root@testserver16 ~]# service mcelogd status
Checking for mcelog
mcelog (pid  32435) is running...
```

For example for Oracle Linux 7:

```
[root@testserver16 ~]# systemctl status mcelogd
Checking for mcelog
mcelog (pid  32435) is running...
```

The status should be "running". If not, it could be stopped or failed.

If mcelog is either not running or failed, the Oracle Linux FMA mce module fails because it requires the mcelog daemon to be working properly for it to function.

2. If the mcelog daemon is running, check the status of the Oracle Linux FMA modules.

   To list the status of all fault manager modules:

```
[root@testserver16 ~]# fmadm config
MODULE                  VERSION STATUS  DESCRIPTION
ext-event-transport     0.2     active  External FM event transport
fmd-self-diagnosis      1.0     active  Fault Manager Self-Diagnosis
ip-transport            1.1     active  IP Transport Agent
mce                        1.0     failed  Machine Check Translator
sysevent-transport      1.0     active  SysEvent Transport Agent
syslog-msgs             1.1     active  Syslog Messaging Agent
```

   In the above example, the mce module has a "failed" status. This means that CPU or memory machine check events are not being monitored by the host and, consequently, not being logged or diagnosed in the fault management database.

3. If the Oracle Linux FMA mce module has failed, confirm the cause of the failure using fmdump.

   For example:

```
[root@testserver16 ~]# fmdump -Ve


May 21 2014 09:56:05.930589483 ereport.fm.fmd.module
nvlist version: 0
   version = 0x0
   class = ereport.fm.fmd.module
   detector = (embedded nvlist)
   nvlist version: 0
      version = 0x1
      scheme = fmd
      authority = (embedded nvlist)
      nvlist version: 0
         version = 0x0
         system-mfg = unknown
         system-name = unknown
         system-part = unknown
         system-serial = unknown
         sys-comp-mfg = unknown
         sys-comp-name = unknown
         sys-comp-part = unknown
         sys-comp-serial = unknown
         server-name = testserver16
         host-id = ffffffff990a7a4a
```

```
        (end authority)

        mod-name = mce
        mod-version = 1.0
    (end detector)

    ena = 0x3631d6cd9f6c0001
    msg = mcelog not running!: client requested that module execution abort
    errno = 1072
    errclass = ereport.fm.fmd.hdl_abort
    __ttl = 0x1
    __tod = 0x52de8a85 0x3777ab2b
```

In the above example, the "`msg =`" field lists that mcelog is not running and is the cause for the mce module failure.

4. Once you have determined that the mcelog daemon is the problem, restart it.

   For example for Oracle Linux 6.5:

   ```
   [root@testserver16 ~]# service mcelogd start
   Starting mcelog daemon
   ```

   For example for Oracle Linux 7:

   ```
   [root@testserver16 ~]# systemctl start mcelogd
   Starting mcelog daemon
   ```

5. Verify that mcelog is running.

   For example for Oracle Linux 6.5:

   ```
   [root@testserver16 ~]# service mcelogd status
   Checking for mcelog
   mcelog (pid  32435) is running...
   ```

   For example for Oracle Linux 7:

   ```
   [root@testserver16 ~]# systemctl status mcelogd
   Checking for mcelog
   mcelog (pid  32435) is running...
   ```

6. Unload the Oracle Linux FMA mce module.

   ```
   [root@testserver16 ~]# fmadm unload mce
   ```

   Doing this generates a fault event that you can identify in the fault management database.

7. Confirm that the unloading of the mce module is captured in the fault management database.

   For example:

   ```
   [root@ban25ts12uut2 ~]# fmadm faulty
   --------------- ------------------------------------  -------------- ---------
   TIME            EVENT-ID                              MSG-ID         SEVERITY
   --------------- ------------------------------------  -------------- ---------
   Jan 21 11:35:07 528fbbb9-92d4-cd7f-ef81-e2fddfd3c244  FMD-8000-2K    Minor

   Problem Status    : solved
   Diag Engine       : fmd-self-diagnosis / 1.0
   System
   ```

```
    Manufacturer  : unknown
    Name          : unknown
    Part_Number   : unknown
    Serial_Number : unknown
    Host_ID       : ffffffff990a7a4a

----------------------------------------
Suspect 1 of 1 :
   Fault class : defect.sunos.fmd.module
   Certainty   : 100%
   Affects     : fmd:///module/mce
   Status      : faulted and taken out of service

Description : A Linux Fault Manager component has experienced an error that
              required the module to be disabled.

Response    : The module has been disabled.  Events destined for the module
              will be saved for manual diagnosis.

Impact      : Automated diagnosis and response for subsequent events
associated
              with this module will not occur.

Action      : Use 'fmadm faulty' to provide a more detailed view of this
event.
              Please refer to the associated reference document at
              http://support.oracle.com/msg/FMD-8000-2K for the latest
service
              procedures and policies regarding this diagnosis.
```

8. Reload the Oracle Linux FMA mce module and confirm that it is running.

   For example:

```
[root@testserver16 ~]# fmadm load /opt/fma/fm/lib/fmd/plugins/mce.so
fmadm: module '/opt/fma/fm/lib/fmd/plugins/mce.so' loaded into fault manager


[root@testserver16 ~]# fmadm config
MODULE                   VERSION STATUS  DESCRIPTION
ext-event-transport      0.2     active  External FM event transport
fmd-self-diagnosis       1.0     active  Fault Manager Self-Diagnosis
ip-transport             1.1     active  IP Transport Agent
mce                      1.0     active  Machine Check Translator
sysevent-transport       1.0     active  SysEvent Transport Agent
syslog-msgs              1.1     active  Syslog Messaging Agent
```

   If the mce module does not unload or reload, restart the fault manager.

   For example for Oracle Linux 6.5:

```
[root@testserver16 ~]# service fmd.init restart
Stopping fmd:                                              [  OK  ]
Starting fmd:                                              [  OK  ]
```

   For example for Oracle Linux 7:

```
[root@testserver16 ~]# systemctl restart fmd.init
Stopping fmd:                                              [  OK  ]
Starting fmd:                                              [  OK  ]
```

Edit mcelog File if Faults Are Not Present in the Fault Management Database

If the entry `raw = yes` in the `mcelog.conf` file is commented out, the Oracle Linux Fault Management software cannot obtain the information it needs to create a fault case. If that happens, fault cases for machine check events processed by mcelog are not added to the fault management database.

1. Confirm that the format of mcelog file messages are in the raw format by opening /var/log/mcelog in a text editor.

   The following table shows an example of a default message and a "raw" message (required by Oracle Linux FMA).

Default Format	Raw Format
Hardware event. This is not a software error. MCE 0 CPU 0 BANK 8 MISC 7 ADDR 102bfc0368 TIME 1383171020 Wed Oct 30 18:10:20 2013 MCG status:EIPV MCIP MCi status: Corrected error Error enabled MCi_MISC register valid MCi_ADDR register valid MCA: MEMORY CONTROLLER RD_CHANNEL0_ERR Transaction: Memory read error STATUS 9c00000000000090 MCGSTATUS 6 MCGCAP 1000c14 APICID 20 SOCKETID 1 CPUID Vendor Intel Family 6 Model 45	CPU 0 BANK 8 TSC 0 RIP 00:0 MISC 0x85 ADDR 0x102bfc0368 STATUS 0x9c00000000000090 MCGSTATUS 0x6 PROCESSOR 0:0x306f1 TIME 1383171020 SOCKETID 1 APICID 20 MCGCAP 0x1000c14

2. If the messages in the mcelog file are in the default format, edit the /etc/mcelog/mcelog.conf file to uncomment the “raw = yes” entry.
3. Delete the old mcelog file that was in the default format.

   ```
   rm /var/log/mcelog
   ```

4. Then restart the mcelog daemon, as follows:

   ```
   service mcelogd restart
   ```

fmd Daemon Might Not Start if SELinux is Running

For the workaround instructions provided in this section, ensure that SELinux tools `sealeart` and `audit2allow`, which are part of the setroubleshoot tool set, are installed.

The fmd daemon might not start if SELinux is running. SELinux protects access to certain directories and files. In particular, access to log files in `/var/opt/fma/fm/fmd` might be denied.

This issue appears when attempting to execute `fmadm` commands. For example, you see the following error:

```
fmadm: failed to connect to fmd: RPC: Program not registered
```

In addition, you can find error messages in the system log like the following:

```
May 28 03:07:14 sca05-0a81e7e6 setroubleshoot: SELinux is preventing
logrotate from read access on the directory /var/opt/fma/fm/fmd. For
complete SELinux messages. run sealert -l 9eb4cb40-9d2b-4428-980f-
c4e46606aec1
```

1. Follow the instructions for running sealert as specified in the log file. For example:

   ```
   sealert -l 9eb4cb40-9d2b-4428-980f-c4e46606aec1
   ```

   The output looks similar to:

   ```
   [root@testserver16 ~]# sealert -l 9eb4cb40-9d2b-4428-980f-c4e46606aec1
   SELinux is preventing logrotate from read access on the
   directory /var/opt/fma/fm/fmd.

   *****  Plugin catchall_labels (83.8 confidence) suggests
   ********************

   If you want to allow logrotate to have read access on the fmd directory
   Then you need to change the label on /var/opt/fma/fm/fmd
   Do
   # semanage fcontext -a -t FILE_TYPE '/var/opt/fma/fm/fmd'
   where FILE_TYPE is one of the following: abrt_var_cache_t, var_lib_t,
   configfile, domain,
   var_log_t, var_run_t, cert_type, configfile, net_conf_t, inotifyfs_t,
   logrotate_t,
   sysctl_kernel_t, mailman_log_t, sysctl_crypto_t, admin_home_t,
   varnishlog_log_t,
   openshift_var_lib_t, user_home_dir_t, var_lock_t, bin_t, device_t, devpts_t,
   locale_t,
   etc_t, tmp_t, usr_t, proc_t, abrt_t, device_t, lib_t, logrotate_var_lib_t,
   root_t,
   etc_t, usr_t, sssd_public_t, sysfs_t, httpd_config_t, logrotate_tmp_t,
   logfile,
   pidfile, named_cache_t, munin_etc_t, mysqld_etc_t, acct_data_t, security_t,
   var_spool_t,
   nscd_var_run_t, sysctl_kernel_t, nfs_t.
   Then execute:
   restorecon -v '/var/opt/fma/fm/fmd'

   *****  Plugin catchall (17.1 confidence) suggests
   **************************

   If you believe that logrotate should be allowed read access on the fmd
   directory by
   default.
   Then you should report this as a bug.
   You can generate a local policy module to allow this access.
   Do
   allow this access for now by executing:
   # grep logrotate /var/log/audit/audit.log | audit2allow -M mypol
   # semodule -i mypol.pp
   ```

2. Execute the following commands as suggested in the log file:

```
grep logrotate /var/log/audit/audit.log | audit2allow -M name
semodule -i name.pp
```

Where *name* is the name of your custom policy module file.

3. Repeat steps 1 and 2 for all the SELinux file access failures. Give different names for each of the .pp files
4. When done, reboot the system.

   Executing `fmadm` commands should now return proper output without a failure message.

Oracle Linux FMA Installation Can Fail When Using Either Anaconda or Oracle System Assistant to Install the OS

Installation of Oracle Linux FMA software can fail when using the Linux Anaconda installer or the Oracle System Assistant assisted OS installation (which calls the Anaconda installer). The OS installation completes successfully and the software is installed; however, upon system reboot after installation the required Oracle Linux FMA services are not automatically started. Consequently, none of the Oracle Linux FMA fault events are recorded or observed on the host.

If this happens, perform the following procedure.

> **Note:**
>
> This procedure only needs to be performed once and does not need to be repeated on subsequent reboots.

1. Complete the OS installation process, ignoring any Oracle Linux FMA software install failure messages.
2. After the system reboots, login as root and make a directory for the man pages.

   ```
   # mkdir -p /usr/local/share/man/man1m
   ```
3. Create soft links to the installed man pages.

   ```
   # ln -s -t /usr/local/share/man/man1m /opt/fma/share/man/man1m/
   fmadm.1m /opt/fma/share/man/man1m/fmdump.1m /opt/fma/share/man/
   man1m/fmd.1m /opt/fma/share/man/man1m/intro.1m
   ```
4. Enable the appropriate services.

   ```
   # chkconfig --add ksyseventd.init
   ```

   ```
   # chkconfig --add fmd.init
   ```
5. Then, start the services.

   ```
   # service ksyseventd.init start
   ```

   ```
   # service fmd.init start
   ```
6. Use the fmadm config command to ensure that all Oracle Linux FMA software components are installed and ready.

   For example:

```
[root@testserver16 ~]# fmadm config

MODULE                   VERSION STATUS  DESCRIPTION
ext-event-transport      0.2     active  External FM event transport
fmd-self-diagnosis       1.0     active  Fault Manager Self-Diagnosis
ip-transport             1.1     active  IP Transport Agent
mce                      1.0     active  Machine Check Translator
sysevent-transport       1.0     active  SysEvent Transport Agent
syslog-msgs              1.1     active  Syslog Messaging Agent
```

Index

M

O

P

R

S

T