# Online System Manager

# (OSM)

# Operations Manual

*PARSEC*$_{\textbf{\textit{Group}}}$

# OSM for OpenVMS

# Operations Manual

**Version 4.2**

PARSEC Group
5610 Ward Rd, Suite 110
Arvada, Colorado 80002
(303) 763-9600
(303) 763-9909 fax

**REVISED:** April 2024
**Copyright © 1994 - 2024 by The PARSEC Group, Inc. All rights reserved**

# Table of Contents



**CHAPTER 1 INTRODUCTION TO OSM 1-1**

**BACKGROUND ..... 1-1**
**OSM STRUCTURE ..... 1-3**
OSM MONITOR ..... 1-3
OSM TRANSPORT ..... 1-3
OSM CONTROL ..... 1-3
**OSM HIGHLIGHTS ..... 1-4**
**OSM FEATURES ..... 1-4**
**OSM MONITORED EVENTS ..... 1-7**
CPU ACCESS MODES ..... 1-7
DEADLOCK DETECTION AND SEARCHES ..... 1-7
DISK FRAGMENTATION ..... 1-7
DISK I/O RATES ..... 1-7
DISK MOUNT VERIFICATION ..... 1-7
FILE SYSTEM CACHES ..... 1-8
FREE DISK SPACE ..... 1-8
GLOBAL PAGE AND SECTION TABLE USAGE ..... 1-8
HARDWARE OR DEVICE ERRORS ..... 1-8
NETWORK NODES ..... 1-8
NETWORK TIME SYNCHRONIZATION ..... 1-9
NONPAGED AND PAGED POOL ..... 1-9
PAGE AND SWAP FILE UTILIZATION ..... 1-9
PAGE FAULTS ..... 1-9
POWER CONDITIONS ..... 1-10
PROCESS AVAILABILITY ..... 1-10
SECURITY ALARMS ..... 1-10
SHARED LOGICAL NAME TABLE USAGE ..... 1-10
SYSTEM CRASHES ..... 1-10
SYSTEM PERFORMANCE ..... 1-11
SYSUAF SYNCHRONIZATION ..... 1-11
**OSM PREREQUISITES ..... 1-12**
HARDWARE REQUIREMENTS ..... 1-12
SOFTWARE REQUIREMENTS ..... 1-12
PRIVILEGE REQUIREMENTS ..... 1-13

**CHAPTER 2 OSM OPERATION 2-1**

**OVERVIEW ..... 2-1**
**OSM_CONTROL IMAGE ..... 2-3**
OSM_CONTROL ..... 2-3
**THE OSM HELP COMMAND ..... 2-4**
HELP ..... 2-4

**CHAPTER 3 OSM PARAMETER FILES AND FORMATS 3-1**

**OVERVIEW ..... 3-1**
PARAMETER FILE FORMAT ..... 3-1
**OSM PARAMETER FORMATS ..... 3-2**
NUMERIC PARAMETERS ..... 3-2
STRING PARAMETERS ..... 3-2
BOOLEAN PARAMETERS ..... 3-2
KEYWORD PARAMETERS ..... 3-3
TIME PARAMETERS ..... 3-3
SEVERITY LEVEL/THRESHOLD PARAMETERS ..... 3-3
**OSM PARAMETER FILE COMMANDS ..... 3-3**
EDIT PARAMETERS ..... 3-3
EXIT ..... 3-5
READ PARAMETERS ..... 3-6
SEND ALL ..... 3-7
SEND REQUEST ..... 3-8
SEND TUNING_INFORMATION ..... 3-9
SET EDITOR ..... 3-10
SHOW EDITOR ..... 3-11
SHOW PARAMETERS ..... 3-12
**PARAMETER FILE PARAMETER ..... 3-13**
EXIT ..... 3-13
**EXAMPLE CONTROL FILE ..... 3-14**

**CHAPTER 4 OSM IDENTIFICATION PARAMETERS 4-1**

**OVERVIEW ..... 4-1**
**IDENTIFICATION COMMANDS ..... 4-2**
SEND CONTACT_INFORMATION ..... 4-2
**CONTACT PARAMETERS ..... 4-3**
ORGANIZATION NAME ..... 4-3
ADDRESS ..... 4-3
CONTACT NAME ..... 4-3
ACCOUNT=NNN ..... 4-3
FAX=NNN-NNN-NNNN ..... 4-3
MODEM=NNN-NNN-NNNN ..... 4-4
PHONE=NNN-NNN-NNNN ..... 4-4
SITE= ..... 4-4
TEXT= ..... 4-4
TEXT_ADD= ..... 4-4

# Table of Contents (Continued)



**CHAPTER 5 CONTROLLING OSM PROCESSES 5-1**

**OVERVIEW ..... 5-1**
STARTING AND STOPPING OSM PROCESSES ..... 5-1
PURGING OSM PROCESSES WORKING SETS ..... 5-2
WAKING THE OSM TRANSPORT PROCESS ..... 5-2
AUTOMATIC STARTUP DURING BOOT ..... 5-3
AUTOMATIC FAILOVER ..... 5-3
AUTOMATIC SHUTDOWN DURING SYSTEM SHUTDOWN ..... 5-4
OSM LOGICAL NAMES ..... 5-4
**MULTI-ENVIRONMENT EXAMPLE ..... 5-8**
NODE DESCRIPTIONS ..... 5-8
NETWORK MONITORING ..... 5-9
OSM TRANSPORT SETUP ..... 5-11
SUMMARY ..... 5-13
**OSM COMMANDS ..... 5-17**
DISABLE AUTOSTART ..... 5-17
ENABLE AUTOSTART ..... 5-18
SHOW STATUS ..... 5-19
START MONITOR ..... 5-20
START TRANSPORT ..... 5-21
STOP MONITOR ..... 5-22
STOP TRANSPORT ..... 5-23
WAKE TRANSPORT ..... 5-24
**ENVIRONMENT PARAMETERS ..... 5-25**
AUTOSTART=300,TRANSPORT ..... 5-25
BURST_DELAY=15 ..... 5-25
CPU_WAKE=300 ..... 5-25
CRITICAL=6 ..... 5-26
EDITOR=TPU ..... 5-26
HEARTBEAT=0 ..... 5-26
LNM_TABLE=LNM$FILE_DEV ..... 5-26
LOG=OPTION ..... 5-26
MONITOR_WAKE=30 ..... 5-28
NIGHT=00:30-05:00 ..... 5-28
OPCOM=N ..... 5-28
OPCOM_ERROR=1 ..... 5-28
PRIME=8:00-17:00,MON,TUE,WED,THU,FRI ..... 5-29
PURGE_WORKING_SET=N ..... 5-29
TRANSPORT_WAKE=300 ..... 5-29
UIC= [1,4] ..... 5-30

**CHAPTER 6 MONITORING FOR EXCEPTION EVENTS 6-1**

**OPERATION ..... 6-1**
**ERRORLOG ENTRY THRESHOLDS ..... 6-2**
**EXCEPTION EVENT MONITORING COMMAND ..... 6-3**
SHOW TEMPERATURE ..... 6-3
**EXCEPTION EVENT MONITORING PARAMETERS ..... 6-4**
AGE_ACCOUNTING=60 00:00:00 ..... 6-5
AGE_ERRORLOG=60 00:00:00 ..... 6-5
AGE_OPERATOR=60 00:00:00 ..... 6-5
ALIGNMENT_FAULTS=2,1000 ..... 6-6
AUDIT=Y ..... 6-7
AUDIT EVENT PARAMETERS ..... 6-7
BUGCHECK=9 ..... 6-8
CAPACITY=1 ..... 6-8
COMPUTABLE=5,95,12,1 ..... 6-9
COMPUTABLE_C=8,98,12,1 ..... 6-9
CONFIGURE=Y, 300 ..... 6-10
CPU=1,20,35,10 ..... 6-10
DEADLOCK=4,1,2 ..... 6-12
DEADLOCK_C=6,2,20 ..... 6-12
DISK=2,95,200 ..... 6-13
DISK_C=7,99,32000 ..... 6-13
ERRFMT=Y ..... 6-14
ERROR TYPE PARAMETERS ..... 6-14
GBLSECTIONS=4,90,80 ..... 6-15
GBLSECTIONS_C=6,97,95 ..... 6-15
IO=3,20,10 ..... 6-16
IO_C=8,32000,32000 ..... 6-16
LOGICAL_NAMES=1,65,0,6 ..... 6-18
PAGE_FAULTS=2,500,40,2 ..... 6-19
PAGEFILE=3,65,70 ..... 6-21
PAGEFILE_C=8,85,90 ..... 6-21
POOL=3,30,65 ..... 6-22
POOL_C=8,85,90 ..... 6-22
POWER=5 ..... 6-23
PROCESSES=5,80,85 ..... 6-24
PROCESSES_C=8,90,95 ..... 6-24
TEMPERATURE=(NO DEFAULT) ..... 6-25
TEMPERATURE_C=(NO DEFAULT) ..... 6-25
XQP_CACHE=3,4,90 ..... 6-26

**CHAPTER 7 USER-DEFINED EVENT MONITORING 7-1**

**OVERVIEW ..... 7-1**
**USER-DEFINED EVENT MONITORING OPERATION ..... 7-1**
**SETTING UP USER-DEFINED EVENT MONITORING ..... 7-2**
USER-DEFINED EVENT MONITORING EXAMPLE ..... 7-4
**USER-DEFINED MONITORING TEMPLATE FILES ..... 7-4**
**MONITORING DISK CONTROLLERS AND TERMINAL SERVERS ..... 7-5**
DS100_DS200_MONITOR_UDM.TEMPLATE ..... 7-5
DS90LPLUS_MONITOR_UDM.TEMPLATE ..... 7-5
DSSI_MONITOR_UDM.TEMPLATE ..... 7-5
**USER-DEFINED EVENT MONITORING COMMANDS ..... 7-6**
CONNECT ..... 7-6
**USER-DEFINED EVENT MONITORING PARAMETERS ..... 7-8**
COMMAND#=SECONDS,FLAGS,COMMAND ..... 7-8

**CHAPTER 8 NETWORK EVENT MONITORING 8-1**

**OVERVIEW ..... 8-1**
**NETWORK EVENT MONITORING OPERATION ..... 8-1**
SETTING UP NETWORK EVENT MONITORING ..... 8-1
**NETWORK EVENT MONITORING COMMANDS ..... 8-5**
SET PASSWORD ..... 8-5
SET TIME ..... 8-6
SHOW TIME ..... 8-7
**NETWORK EVENT MONITORING: PARAMETERS ..... 8-8**
DECNET_OBJECT=0 ..... 8-8
DEFAULT_NETWORK=DECNET ..... 8-8
NETWORK#=SEVERITY,SECONDS,NODE-NAME,NETWORK ..... 8-9
NETWORK_TIME#=SEVERITY,SECONDS,NODE-NAME,DELTA-TIME,NETWORK ..... 8-9
NETWORK_TIME_SET#=NODE-NAME,NETWORK ..... 8-10
NETWORK_ASCII=0 ..... 8-10
NETWORK_ID=0 ..... 8-10
TCPIP_PORT=0,0 ..... 8-11
TIME_SET=300,0 0:0:15 ..... 8-11

**CHAPTER 9 EXCEPTION EVENT REPORTING 9-1**

**OVERVIEW ..... 9-1**
**CONTROLLING OUTPUT IN ASCII FILES ..... 9-2**
CONTROLLING REPEATED EXCEPTION EVENTS IN ASCII FILES ..... 9-2
CONTROLLING THE AMOUNT OF DETAIL IN ASCII FILES ..... 9-3
**OUTPUT GROUPS ..... 9-4**
**TRANSPORT PROCESS REPORTING PARAMETERS ..... 9-6**
ACTIVE=0:00-0:00,SUN,MON,TUE,WED,THU,FRI,SAT ..... 9-6
ANALYZE/ERROR= ..... 9-6
BINARY=FILE-SPEC ..... 9-7
CATEGORY=ALL ..... 9-8
DETAIL=SEVERITY ..... 9-9
DETAIL_ERRORLOG=ALL ..... 9-11
DIAGNOSE=/SUMMARY ..... 9-11
DISABLED= ..... 9-11
ELV=/SUMMARY ..... 9-12
ENABLED= ..... 9-12
ERRORLOG_FORMAT=OSM ..... 9-13
ERRORLOG_TRANSLATE=MONITOR ..... 9-13
ERRORLOG=OSM$TPG:%04D%02D%02D%02D.DAT ..... 9-13
FILE=OSM$:.OSM_ASCII; ..... 9-14
HBOUTPUT=MAIL ..... 9-14
MAIL= ..... 9-14
OPERATOR=CENTRAL ..... 9-15
OUTPUT=MAIL ..... 9-16
OUTPUT_ACCOUNTS= ..... 9-17
OUTPUT_REASONS=ALL ..... 9-18
OUTPUT_SEVERITY=0 ..... 9-19
OUTPUT_SEVERITY_MAX=10 ..... 9-19
PRINTER= ..... 9-19
REPEAT_ALWAYS=Y ..... 9-19
REPEAT_DOM=0 ..... 9-19
REPEAT_DOW=0 ..... 9-19
SCRIPT= ..... 9-20
TEXT_FORMAT=BRIEF ..... 9-20
TEXT_MODE=FILE ..... 9-21
TEXT_PREFIX= ..... 9-21
TEXT_PREFIX_ADD= ..... 9-21
TEXT_POSTFIX= ..... 9-21
TEXT_POSTFIX_ADD= ..... 9-21

**CHAPTER 10 OSM PAGING SUPPORT 10-1**

**OVERVIEW ..... 10-1**
**OPERATION ..... 10-2**
GENERATING AND INVOKING A PAGE ..... 10-2
PAGING NOTIFICATION STRING ..... 10-3
PAGING FOR ERRORLOG ENTRIES ..... 10-3
REPEAT PAGING ..... 10-4
BACKUP PAGING ..... 10-5
PAGING HISTORY ..... 10-5
**OSM PAGING COMMANDS ..... 10-6**
CLEAR PAGER ..... 10-6
DELETE PAGER ..... 10-7
SHOW PAGER ..... 10-8
**OSM PARAMETERS RELATED TO PAGING ..... 10-9**
DETAIL ..... 10-9
OUTPUT ..... 10-9
**GENERAL PAGER CONTROL PARAMETERS ..... 10-10**
PAGER=0 ..... 10-10
PAGER_DELETE=14 0:0:0 ..... 10-10
PAGER_RETRY=1,0 00:00:00 ..... 10-10
**EXCEPTION EVENT PAGING PARAMETERS ..... 10-11**
BUGCHECK_P=PAGER-STRING ..... 10-11
CAPACITY_P=PAGER-STRING ..... 10-11
COMMAND#_P=PAGER-STRING ..... 10-11
COMPUTABLE_P=PAGER-STRING ..... 10-11
CPU_P=PAGER-STRING ..... 10-12
DEADLOCK_P=PAGER-STRING ..... 10-12
DISK_P=PAGER-STRING ..... 10-12
GBLSECTIONS_P=PAGER-STRING ..... 10-12
IO_P=PAGER-STRING ..... 10-12
LOGICAL_NAMES_P=PAGER-STRING ..... 10-12
NETWORK_ASCII_P=PAGER-STRING ..... 10-12
NETWORK#_P=PAGER-STRING ..... 10-12
NETWORK_TIME#_P=PAGER-STRING ..... 10-13
OPCOM_ERROR_P=PAGER-STRING ..... 10-13
PAGEFILE_P=PAGER-STRING ..... 10-13
PAGE_FAULTS_P=PAGER-STRING ..... 10-13
POOL_P=PAGER-STRING ..... 10-13
POWER_P=PAGER-STRING ..... 10-13
PROCESSES_P=PAGER-STRING ..... 10-13
TEMPERATURE_P=PAGER-STRING ..... 10-13
XQP_CACHE_P=PAGER-STRING ..... 10-13
ERROR FORMAT PAGING PARAMETERS ..... 10-14
AUDIT SERVER PAGING PARAMETERS ..... 10-15

**CHAPTER 11 AUTOMATED FIXES 11-1**

**OVERVIEW ..... 11-1**
AUTOMATED FIXES FOR ERRORLOG ENTRIES ..... 11-3
**ENABLING AN AUTOMATED FIX AUDIT TRAIL ..... 11-3**
**AUTOMATED FIX TEMPLATE FILES ..... 11-3**
**AUTOMATED FIX PARAMETERS ..... 11-4**
ALIGNMENT_FAULTS_F=FLAGS,COMMAND ..... 11-4
BUGCHECK_F=FLAGS,COMMAND ..... 11-4
CAPACITY_F=FLAGS,COMMAND ..... 11-4
COMMAND#_F=FLAGS,COMMAND ..... 11-4
COMPUTABLE_F=FLAGS,COMMAND ..... 11-4
COMPUATABLE_CF=FLAGS,COMMAND ..... 11-4
CPU_F=FLAGS,COMMAND ..... 11-5
DEADLOCK_F=FLAGS,COMMAND ..... 11-5
DEADLOCK_CF=FLAGS,COMMAND ..... 11-5
DISK_F=FLAGS,COMMAND ..... 11-5
DISK_CF=FLAGS,COMMAND ..... 11-5
GBLSECTIONS_F=FLAGS,COMMAND ..... 11-5
GBLSECTIONS_CF=FLAGS,COMMAND ..... 11-5
IO_F=FLAGS,COMMAND ..... 11-5
IO_CF=FLAGS,COMMAND ..... 11-5
LOGICAL_NAMES_F=FLAGS,COMMAND ..... 11-5
NETWORK_ASCII_F=FLAGS,COMMAND ..... 11-5
NETWORK#_F=FLAGS,COMMAND ..... 11-6
NETWORK_TIME#_F=FLAGS,COMMAND ..... 11-6
PAGEFILE_F=FLAGS,COMMAND ..... 11-6
PAGEFILE_CF=FLAGS,COMMAND ..... 11-6
PAGE_FAULTS_F=FLAGS,COMMAND ..... 11-6
POOL_F=FLAGS,COMMAND ..... 11-6
POOL_CF=FLAGS,COMMAND ..... 11-6
POWER_F=FLAGS,COMMAND ..... 11-6
PROCESSES_F=FLAGS,COMMAND ..... 11-6
PROCESSES_CF=FLAGS,COMMAND ..... 11-6
TEMPERATURE_F=FLAGS,COMMAND ..... 11-6
TEMPERATURE_CF=FLAGS,COMMAND ..... 11-6
XQP_CACHE_F=FLAGS,COMMAND ..... 11-7
ERROR FORMAT AUTOMATED FIX PARAMETERS ..... 11-7
AUDIT SERVER AUTOMATED FIX PARAMETERS ..... 11-8

**CHAPTER 12 SYSUAF SYNCHRONIZATION 12-1**

**OVERVIEW ..... 12-1**
**FEATURES ..... 12-2**
**RESTRICTIONS ..... 12-2**
**COMPONENT INTERACTION ..... 12-3**
**DISTRIBUTION LIST ..... 12-3**
EXAMPLE 1 ..... 12-5
EXAMPLE 2 ..... 12-5
EXAMPLE 3 ..... 12-6
EXAMPLE 4 ..... 12-6
EXAMPLE 5 ..... 12-7
**LOGICAL NAMES ..... 12-9**
OSM_SYSUAF_CLASS ..... 12-9
OSM_SYSUAF_CLASS_NODENAME ..... 12-9
OSM_SYSUAF_FIELD_CLASS_N ..... 12-10
OSM_SYSUAF_USER_CLASS_N ..... 12-11
**SYSUAF SYNCHRONIZATION COMMANDS ..... 12-12**
CLEAR SYSUAF ..... 12-12
SHOW STATUS ..... 12-12
SYNCHRONIZE ..... 12-13
**SYSUAF SYNCHRONIZATION PARAMETERS ..... 12-17**
SYSUAF=FILE-SPECIFICATION ..... 12-17
SYSUAF_ALLOW=OPTIONS ..... 12-17
SYSUAF_DISTRIBUTE=NODENAME,... ..... 12-18
SYSUAF_DISTRIBUTE_ADD=NODENAME,... ..... 12-18
SYSUAF_DISTRIBUTE_VERIFY=0,NODENAME,... ..... 12-18
SYSUAF_DISTRIBUTE_VERIFY_ADD=NODENAME,... ..... 12-18
SYSUAF_NODE=NODENAME ..... 12-18
SYSUAF_NOREMOVE=SYSTEM,DEFAULT ..... 12-18
SYSUAF_NOTIFY=NODENAME... ..... 12-18

**CHAPTER 13 OSM TESTING 13-1**

**OVERVIEW ..... 13-1**
**DEMONSTRATE COMMANDS ..... 13-2**
DEMONSTRATE ALIGNMENT_FAULTS ..... 13-2
DEMONSTRATE AUDIT_SERVER ..... 13-3
DEMONSTRATE BREAKIN ..... 13-4
DEMONSTRATE BUGCHECK ..... 13-5
DEMONSTRATE CPU ..... 13-6
DEMONSTRATE DEAD_LOCKS ..... 13-7
DEMONSTRATE DISK DEV-NAME ..... 13-8
DEMONSTRATE ERRORLOG FILE-NAME ..... 13-9
DEMONSTRATE GLOBAL_SECTIONS ..... 13-10
DEMONSTRATE IO ..... 13-11
DEMONSTRATE LOGICAL_NAMES ..... 13-12
DEMONSTRATE NETWORK_OBJECT ..... 13-13
DEMONSTRATE PAGE_FAULTS ..... 13-14
DEMONSTRATE PAGR ..... 13-15
DEMONSTRATE XQP_CACHE ..... 13-16

**CHAPTER 14 MONTHLY SUMMARY REPORTING 14-1**

**OVERVIEW ..... 14-1**
OSM_SUMMARY_SETUP.COM ..... 14-1
OSM_SUMMARY_UDM.TEMPLATE ..... 14-2
**SETTING UP MONTHLY SUMMARY REPORTING ..... 14-2**
**OSM SUMMARY COMMANDS ..... 14-4**
SUMMARIZE/OSM ..... 14-4
SUMMARIZE/ERRLOG ..... 14-6
**SUMMARY PARAMETERS ..... 14-7**
ENHANCE= ..... 14-7
NORMAL= ..... 14-7
**EXAMPLE SUMMARY REPORT ..... 14-8**

**CHAPTER 15 OSM INSTALLATION 15-1**

**OVERVIEW ..... 15-1**
INSTALLING OSM IN A VMSCLUSTER ENVIRONMENT ..... 15-1
**UPDATING OSM ..... 15-1**
**OPENVMS OSM INSTALLATION EXAMPLE ..... 15-2**
**INSTALLING OSM ON ADDITIONAL VMSCLUSTER NODE(S) ..... 15-10**
**UPDATING OSM ON ADDITIONAL OPENVMS DECNET NODE(S) ..... 15-10**
**LICENSE INSTALLATION INFORMATION ..... 15-10**

**APPENDIX A PARAMETER LIST A-1**

**APPENDIX B OSM FILE SUMMARY B-1**

**PERMANENT FILES ..... B-1**
AAAAAA.DIR ..... B-1
BINARY.DIR ..... B-1
ERRLOG_ALPHA.SYS ..... B-1
ERRLOG_I64.SYS ..... B-1
ERRLOG_VAX.SYS ..... B-1
LINK.COM ..... B-1
OSM_AXP_YYYY.EXE ..... B-2
OSM_IA64_YYYY.EXE ..... B-2
OSM_VAX_YYYY.EXE ..... B-2
OSM.LOG ..... B-2
OSM.THRESHOLDS ..... B-2
OSM_ADDNODE.COM ..... B-2
OSM_CONTROL_AXP_YYYY.EXE ..... B-2
OSM_CONTROL_IA64_YYYY.EXE ..... B-2
OSM_CONTROL_VAX_YYYY.EXE ..... B-2
OSM_CTLFILE.DAT ..... B-2
OSM_CTLFILE_AAAAAA.DAT ..... B-3
OSM_LIB.OLB ..... B-3
OSM_LIB_IA64.OLB ..... B-3
OSM_LIB_AXP.OLB ..... B-3
OSM_LOGICALS.COM ..... B-3
OSM_LOGICALS.TEMPLATE ..... B-3
OSM_GETDEVICE_INFO.COM ..... B-3
OSM_HELP.HLB ..... B-3
OSM_MON_START.COM ..... B-3
OSM_NODE_SETUP.DAT ..... B-3
OSM_INFO.DAT ..... B-4
OSM_INFO.FDL ..... B-4
OSM_PARAMS.TRANSPORT ..... B-4
OSM_PARAMS_*.MONITOR ..... B-4
OSM_REMOVE.COM ..... B-4
OSM_SDA.COM ..... B-4
OSM_SDA_COMMANDS.COM ..... B-4
OSM_SHUTDOWN.COM ..... B-4
OSM_STARTUP.COM ..... B-5
OSM_STARTUP.TEMPLATE ..... B-5
OSM_TRANSPORT_AXP_YYYY.EXE ..... B-5
OSM_TRANSPORT_IA64_YYYY.EXE ..... B-5
OSM_TRANSPORT_VAX_YYYY.EXE ..... B-5
OSM_TRANSPORT.LOG ..... B-5
OSM_UPDATE.COM ..... B-5
OSM_XPORT_START.COM ..... B-5
OUTPUT_*.TEMPLATE ..... B-5
TRANSPORT.REPEAT ..... B-5
WORK ..... B-5
*_UDM.TEMPLATE ..... B-6
*_FIX.TEMPLATE ..... B-6

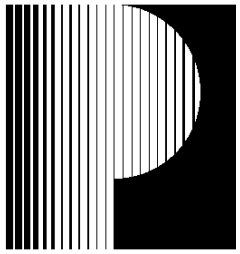
**TEMPORARY FILES** ..... **B-6**
OSM_COMMAND*.* ..... B-6
OSM_CRASH.DAT ..... B-6
OSM_DATAFILE_AAAAAA.ACTIV ..... B-6
OSM_DATAFILE_AAAAAA.ACTIVE ..... B-6
OSM_DATAFILE_AAAAAA.SEND ..... B-7
OSM_INFORMATION.DAT ..... B-7
OSM_INFORMATION.DEVICE ..... B-7
OSM_INFORMATION.SYSGEN ..... B-7
**HANDLING TEMPORARY FILES** ..... **B-7**

**INDEX** ..... **1**

# OSM

Quick Start Guide
For
OpenVMS

THE PARSEC GROUP



## Installation

Install OSM with the command @sys$update:vmsinstal OSM ddcu: where ddcu is the name of the tape drive on which the OSM media has been mounted.

## Starting

Start OSM with the command @sys$manager:osm_startup.

## Automatic Startup

To automatically restart OSM each time the system is rebooted, add the startup command to the system wide startup procedure, named sys$manager:systartup_vms.com.

## Foreign Command

OSM is invoked using a foreign command. This command is normally created in either the system wide login procedure or user specific login procedures. The command to create the foreign command is OSM:==$OSM_CONTROL.

## Stopping

OSM may be completely stopped with the commands

OSM STOP MONITOR

OSM STOP TRANSPORT

## Editing Parameters

Edit cluster command parameters with the command

OSM EDIT PARAMETERS.

Edit node specific parameters with the command OSM EDIT PARAMETERS/NODE=name.

## Deinstallation

Remove OSM from the system with the command

@osm$:osm_remove

## Technical Support

For assistance with OSM, please call

PARSEC Group

Phone: (888) 4-PARSEC

(888) 472-7732

http://www.parsec.com/

# Chapter 1
# Introduction to OSM

## Background

Most data processing departments are increasing the number of systems and types of systems they need to perform company tasks while decreasing the number of support personnel. System managers are being required to be more knowledgeable about these different systems, and are required to handle all the problems that may arise. Systems are no longer in a central location; instead, they exist throughout the entire office building and in many cases, throughout the country or world. Typically, system managers have little or no time to proactively manage the systems; more often than not, they are in firefighting mode. They certainly do not have the time to log onto each system and look around to determine if these systems are functioning properly.

The Online System Manager (OSM) software product aids in the management, predictive maintenance, and support of OpenVMS. OSM allows system administrators to automatically monitor hardware, software, resource, performance, security, and user-defined events that represent potential system failures. Once OSM detects an event, it reports that event to a specified parties normally via E-mail. This allows corrective action before the event becomes a failure. OSM can execute a user-supplied command procedure, batch file or DCL command to handle the event.

OSM allows the user to associate a priority level from one through ten to every monitored event. Priority 1 is the lowest and priority 10 is the highest. The user can also define which of these priorities are critical and which are not. By default, critical events are all events that occur at priority six and above.

When OSM detects an event, it first determines if the event's priority is critical. If it is, OSM immediately sends the information collected about the event to the specified party or parties. If the event is non-critical, OSM will store the event information and send it to the specified party once a night. For all events, both critical and non-critical, OSM can also generate any page and invoke any user-defined fix for the problem, if either of these is specified.

OSM is VMScluster-compatible. In a VMScluster environment, only the monitoring process needs to run on every node in the cluster. The OSM transport process runs on a single node of the cluster. This allows the product to concentrate all the data in one location and send the information to the system manager, PARSEC Group, your Customer Dispatch Center, or any other designated party. The executable images and data files for the VAX, Alpha and Integrity server environments are maintained in the same directory.

OSM is fully configurable through the use of parameter files and logical names. These parameter files are maintained through the use of any text editor on the system. The parameters

allow the user to control all aspects of OSM operation. For instance, there are parameters that allow the user to specify where OSM sends event information, what the individual event priorities are, which events are critical, how to handle an event (called automated fix), what telephone number to page, event thresholds and much more.

OSM parameter files contain the user-defined parameters that control OSM. In a VMScluster, there is one cluster-wide parameter file, supplemented with node-specific parameter files. These parameter files control both the monitor and transport processes, which are described later. OSM gets its parameters by first examining the cluster-wide file, then the node-specific file; allowing the node-specific file to override cluster-wide values for a particular node.

The monitor process uses these files to determine thresholds for data collection and notification. The transport process uses these files to determine where to send the information when the monitor process finds a reportable condition. The transport process is capable of sending this data to multiple destinations. For example, it can send hardware error reports to your hardware service vendor, as well as a copy to the system manager.

# OSM Structure

OSM is composed of four separate software components: the OSM Monitor, OSM Transport, and OSM Server processes and the OSM Control image. Not all of these processes will be running on all systems being monitored by OSM.

## OSM Monitor

This process runs on every node for which OSM is licensed. It is responsible for monitoring the system and looking for performance events, security events, depletion of disk space, user-defined events, system crashes and hardware and software events. If OSM detects any of its monitored events, it logs the information to a file on disk (a .SEND file) which is visible to the transport process.

If a command, command procedure or batch file should be executed upon detection of one of the monitored events to fix the problem, the monitor process creates a subprocess to execute the user-specified fix procedure or command. The monitor also generates a page if it detects an exception event that the user has specified should generate a page, allowing immediate notification of the exception event.

## OSM Transport

This process wakes up every user-defined interval looking for one of the files recorded by a monitor process. These files are referred to as .SEND files, since the monitor process puts the files that need to be sent in the OSM directory with a file type of ".SEND". In a cluster, only one transport process is needed. Once it finds a .SEND file, the transport process sends the information in the file to user-defined locations using user-defined mechanisms. For example, the .SEND file can be translated to ASCII and sent to the system manager via E-mail, it may be sent as in abbreviated message to a cell phone or pager, or the binary .SEND file could be sent over DECnet to a centralized location where the management team resides.

The transport process is also responsible for determining if a page is associated with an event and initiating that page. If a page is associated with an event that the monitor process has reported, the transport process invokes the page by sending the appropriate mail messages.

## OSM Control

OSM Control is an image, not a process. It is invoked by issuing the OSM command at the DCL prompt on OpenVMS. This is the user interface for OSM. With OSM Control, you can start and stop OSM processes, edit OSM parameter files, cause any .SEND files to be sent, control the paging function and control overall OSM functionality.

# OSM Highlights

- Allows system administrators to manage a larger number of systems through a "Management by Exception" policy
- Proactive maintenance approach to hardware and software support
- Automatic "call home" feature
- Flexible event reporting, including electronic mail, networks, and printers
- Ability to page a pager upon detection of an exception event
- Ability to implement a user-defined fix upon detection of an exception event
- Ability to incorporate user-defined exception monitoring
- Daily system status summary reporting
- Software and hardware events automatically dispatched to customer-specified location(s)
- Ability to dispatch hardware and software calls to different locations
- User-definable event thresholds
- User-definable severity levels
- Full VMScluster support
- Works with OpenVMS Integrity, OpenVMS Alpha, and OpenVMS VAX
- Network monitoring
- SYSUAF synchronization
- Ease of use

# OSM Features

- **Management by Exception**
  OSM provides a proactive maintenance approach to hardware and software events. With OSM you no longer have to log on to each system to determine if there is any potential or existing problems. OSM will continually monitor your system and report on exceptions to a location of your choosing.

- **Automatic Notification**
  Typically, system managers do not have the time to log into each system for which they are responsible to determine whether the system is functioning satisfactorily. With this feature, the system manager will automatically be notified of any exceptions to normal system behavior, indicating that they need to address a potential, or existing, problem on the system(s) on which OSM has detected an exception.

- **Automated Fixes**
  OSM provides the user with the ability to specify any command, command procedure or batch procedure to be invoked when it detects either a critical or non-critical event. With this feature, OSM can detect and correct exception events without any human intervention, freeing up the system manager's time for other tasks.

- **Paging**
  If enabled, OSM will automatically page the specified party when it detects an event. This allows the system manager to be notified immediately of events that may cause problems without having to be physically at the site, or even logged into the system. If needed, OSM may also be set up to page a backup pager in the event the primary party is unable to respond to their page.

- **User-Defined Event Monitoring**
  OSM monitors for a vast number of exception events. These events can cause anything from minor user inconveniences to catastrophic system failures. However, OSM could never know about exception events that are specific to a site or an application. Therefore, OSM provides the ability to monitor user-defined events. OSM accomplishes this by executing a user-supplied procedure that creates a data file in a known directory. When OSM discovers this data file, OSM will handle the user-defined event exception information the same way it handles the event information it collects. For example, it can call a fix procedure, page someone or simply send the specified parties E-mail on the problem.

- **Autostart and Failover for the OSM Transport Process**
  The OSM transport process only needs to run on one node in a VMScluster. The transport process can automatically fail over to another node in the cluster if the node on which it is running goes down to ensure that it is always available. To facilitate this, the monitor process periodically checks to see if the transport process is running. If it is not, the monitor process automatically starts the transport process. The user can control which nodes of the cluster that may run the OSM transport process.

- **Automatic "call home" feature**
  When OSM detects an exception event, it will automatically send the information to a location that you have specified. For example, if there is one system to which you are always connected, OSM can report on all exceptions that occur on any remotely monitored system to your system. This will save you a vast amount of time hunting for potential or existing problems.

- **Flexible event reporting, including E-mail and networks**
  You can choose how you want exception information detected on remote nodes reported to you. If the remote node is connected to your system via a network, OSM can transfer data over the network. OSM can also transfer data over the Internet using SMTP or other compatible mail interface.

- **Daily system status summary reporting**
  Once a day, during a user-defined interval of time, OSM sends non-critical exception information to the locations specified by the user.

- **Software and hardware events automatically dispatched to customer-specified location(s)**
  You have the ability to define exceptions that you feel are critical, and exceptions that you feel are not critical. When OSM detects a critical event, information about that event is immediately sent to the specified customer location. If the events are non-critical, OSM saves the event information in an event file and sends the event information once each night to the customer-specified location.

- **Ability to dispatch hardware and software calls to different locations**
  Often times it is advantageous to send only the hardware events to a different location. For example, you might have different people performing your hardware support and your software support. OSM allows you to separate this information and send hardware events to a specific location and all events (including hardware) to a central location.

- **User-definable event thresholds**
  Not enough data can hurt you, but too much data can also be detrimental to effective system management. Therefore, OSM allows you to specify thresholds for exception events. OSM will not report on an exception event unless it exceeds the user-defined threshold for that event.

- **User-definable severity levels**
  Every system is different. Certain exception events might be very important to you while those same events might not mean anything to someone else. Therefore, OSM allows you to define the severity levels of each event. Severity levels can be from 0 through 10; 10 being the most critical, and 0 disabling the event detection and actions.

- **Full VMScluster support**
  OSM was designed to run in either a cluster environment or on a standalone system. In a cluster environment, you can share parameter files, transport processes, as well as several other OSM resources.

- **Network monitoring**
  OSM can monitor OSM on other nodes of the network. This allows you to be notified when another node of the network becomes inaccessible or is not being monitored.

- **SYSUAF synchronization**
  OSM monitors the audit server process for notification of security events. When one of those events indicates a modification to the SYSUAF, OSM can pass that information to OSM on other nodes of the network. Through this process, those changes can be selectively applied to other systems. For example, each time a user changes their password, that can be propagated to other systems to keep them in sync.

- **Ease of use**
  OSM is very flexible. You may modify parameters controlling just about anything OSM does. However, too much flexibility can sometimes make it difficult to set up an application. OSM eliminates this by having default values for all parameters. This allows you to start benefiting from the use of OSM immediately upon installation.

# OSM Monitored Events

Each monitored event has a user-defined threshold that allows the system manager to specify what normal behavior is, and what is an exception to normal behavior. Along with the user-defined threshold, the system manager can determine what level of severity to assign to each of the exceptions, and which of those levels are critical.

If one of the monitored events exceeds the specified threshold, called an exception event, OSM collects and records the appropriate information. If the exception event is critical, all the information about the exception event is immediately sent to the user-defined location. When all the exception events are non-critical, they are stored and sent to the system manager-defined location, at a system-defined time each day; the default being shortly after midnight. Exception event information can be sent to the system manager, a central system management team, or anywhere that is accessible via E-mail or a network. The following list summarizes the events that OSM can monitor.

## CPU Access Modes

OSM monitors the amount of time spent in each of the CPU access modes. This information can be used to determine if you are running out of CPU capacity and whether the system is spending too much time in overhead activities as opposed to user activities.

## Deadlock Detection and Searches

A deadlock can occur when two or more processes are waiting on each other's resources. Deadlocks can cause very poor performance and application failures. Typically, they are caused by application design. There are system parameters that can help if you know you have a problem. OSM monitors the system for both deadlocks and deadlock searches, and notifies the specified party if either of these exceeds the specified threshold.

## Disk Fragmentation

Excessive disk fragmentation forces the system to do multiple physical I/Os to obtain a single piece of disk information (called a split I/O), or is forced to update the file mapping pointers in memory (called a window turn). Disk I/Os are currently the slowest part of normal system operation. Avoiding additional I/Os can greatly increase system performance. OSM monitors both categories of disk fragmentation: split I/Os and window turns. This information indicates that you either need to perform a disk backup and restore procedure, or consider purchasing a disk defragmentation product to maintain contiguous space and files.

## Disk I/O Rates

I/O saturation can occur when too many I/Os are sent to the same disk. If your I/O load is improperly balanced, a high I/O rate could be the indication. OSM monitors and reports on excessive I/O rates, giving you an indication of possible I/O saturation.

## Disk Mount Verification

When OpenVMS detects a problem accessing a mounted disk, it puts the disk in a mount verification state. During this time, all attempts to do user I/O to that disk will be stalled and

the application will hang. When everything is OK again, the mount verification will complete and user I/O requests will resume. Many applications can not accept the delays caused by mount verification. Either user or hardware errors cause most mount verifications. OSM can report when a disk enters mount verification, thereby speeding the process to resolving the problem.

## File System Caches

A low file system cache hit rate can cause additional unneeded I/Os. I/Os are the bane of system performance, and unnecessary I/Os are even worse. OSM detects when a system has a low file cache hit rate with a high attempt rate. This allows you to alleviate the performance problem by increasing the appropriate system parameters.

## Free Disk Space

If you run out of space on the system disk, your system can hang. If you run out of space on other disks, your applications may quit working properly. OSM continually monitors the available disk space to prevent system hangs or application failures due to insufficient disk resources. OSM checks for a percentage of free space on a volume-by-volume basis. A volume that drops below the specified free space will cause OSM to notify the specified parties. The percentage values are user-definable either system-wide or on a per volume basis. This allows the system manager to define different values for critical versus non-critical volumes.

## Global Page and Section Table Usage

Global page table entries and global section table entries are needed to install sharable known images on OpenVMS. Either not having enough global pages or global sections, or having a fragmented global page table will cause problems when trying to install something shared. The solution to these problems is typically modifying the global page and section table system parameters and rebooting the system. That is not a problem if you know that you are running out of these resources, because you can schedule a time to fix the problem. However, if you do not know and you need to install something now, you have a major problem. OSM will report on global page and section table deficiencies when they occur so you can schedule time to fix them.

## Hardware or Device Errors

Hardware errors can often lead to increased system downtime. You need to know when errors are occurring so that you can rectify the situation. Having OSM automatically and continuously monitoring for device and system hardware errors eliminates the need for the system manager to manually search through the error log file. OSM will notify the specified parties of any errors that are problematic. OSM monitors these device errors through the OpenVMS ERRFMT process.

## Network Nodes

OSM may be set up to have one node of the network to monitor other nodes on the network. When a system goes down; no software, no matter how good, can report that the system on which it was running is down. By allowing each node to verify that OSM is running on other

nodes, it is possible to always be sure that everything is being monitored properly. OSM supports both DECnet and TCP/IP for its network communication.

## Network Time Synchronization

Some applications, especially distributed databases, work best when the time on each node is the same. OSM has the ability to poll other nodes of the network and report when the time on that node is not the same as it is on the local node. It also has the ability to set the time on the local node to the time reported by the remote node.

## Nonpaged and Paged Pool

Nonpaged pool is a critical system resource and if undersized, can have a devastating effect on system performance and potentially cause the system to hang. OSM monitors and reports on non-paged pool expansion, paged pool usage and allocation failures. This allows a system manager to schedule a time for modification of these system parameters if they are too low. You can then reboot during off-hours instead of possibly having the system hang during prime time and having to deal with frustrated users.

## Page and Swap File Utilization

OpenVMS monitors page and swap file utilization. If it exceeds an OpenVMS-determined threshold, it will send a single message to the console terminal. The information is not logged to any other location and, if not resolved, will typically cause the operating system to hang. Since this information only goes to the system console once, your chance of seeing the message and fixing the problem before the system hangs is small.

OSM automatically monitors this information as well. If utilization exceeds the user-definable threshold, it will alert all specified parties of the pending problem greatly increasing your chance for alleviating the problem before it causes your operating system to hang. With OSM, it is not necessary to have a person constantly monitoring the console terminal for page and swap file problems.

## Page Faults

Page faults are a normal fact of life on virtual memory systems such as OpenVMS. But excessive page faulting is an indication of either a poorly tuned system, or a system that needs more resources. OSM monitors and reports on the system-wide page fault rate, the system-wide hard fault rate, and the operating system fault rate. If any of these exceeds the user-defined threshold, OSM automatically reports to the specified parties.

## Power Conditions

It is essential that you know when your systems are up or down and when they have either crashed or gone through a successful power fail recovery. The problem is that too many systems scattered around your facility make it impractical for you to continually check on them. OSM alleviates this problem by automatically reporting when a system successfully completes a power fail recovery.

## Process Availability

OSM monitors system limits for the number of processes and the number of interactive jobs. You can be notified before one of these limits is reached so you can still log in to correct the issue.

## Security Alarms

OSM receives security messages directly from the OpenVMS security software. It then identifies the type of security event and immediately notifies the specified parties of those that are considered serious.

## Shared Logical Name Table Usage

Having an incorrectly sized logical name hash table can be detrimental to system performance. OSM will determine when there is a potential problem with the shared logical name hash table. OSM monitors and reports on the number of entries in the longest logical name hash table entry, the percentage of the logical name hash table entries that are used, and the average number of entries in each of the used logical name hash table entries.

## System Crashes

System crashes happen for a reason that is often detrimental to the systems operation. If a system crashes due to hardware or software problems, you should know about it because chances are, it will happen again. Upon booting an OpenVMS system, OSM can check to see if the boot was a result of a system crash and if so, OSM dispatches the appropriate information to the designated location. This feature requires the system boot procedures be modified to call an OSM specific command procedure to collect this information.

## System Performance

In today's complex computing environments, system managers are often responsible for the performance of a number of systems. Typically, performance management is at the bottom of their list of priorities. There is a decent selection of very good tools that are available to analyze system performance and make recommendations on how to improve that performance. The only problem is that with most of these, you must first log onto each system and run the performance analysis product just to determine if there is a problem. Then you must analyze the data to determine how to fix the problem.

OSM works hand in hand with performance analysis tools. It is complementary rather than competitive to performance analysis products. Utilizing OSM, you can eliminate the first step, which is determining which node has a potential performance problem. By setting thresholds on each of the performance categories, you can have OSM running on each of your systems, continually searching for performance bottlenecks. If it detects one, it will notify you of the condition. At that point, you can utilize any performance analysis product to determine how to alleviate the bottleneck. With OSM, you do not have to continually search for the systems with performance problems; they will be automatically reported to you freeing up your time for other tasks.

## SYSUAF Synchronization

In a network environment, there are generally some users, such as the system managers who have an account on several different computers. Frequently, the password for each of these accounts must be changed at different times. To keep track of their passwords, these users usually resort to writing down their password on paper, or spending time logging in to each computer and changing the password when they are forced to change it on one node. Neither of these are desired options. To alleviate this, OSM can propagate changes to a user’s account on one node of the network to several other nodes. This feature may also be used to propagate changes to other fields of each user’s record in the system authorization file.

# OSM Prerequisites

## Hardware Requirements

The following list provides the user with the minimal hardware configuration required to run OSM. Additional disk or memory may be required on some systems in a large environment to support storing reported events for several computers or for a large period of time.

- HP VAX, Alpha, or Integrity system running HP or VSI OpenVMS
- Approximately 20MB of disk space for installation and use

## Software Requirements

For VAX systems:

- OpenVMS VAX V6.0 or higher
- VAXCRTL.EXE is part of OpenVMS VAX, but may have been tailored off. This file must be installed for OSM to work.

For Alpha Systems:

- OpenVMS Alpha V6.1 or higher

For Integrity Server Systems:

- OpenVMS IA64 V8.2 or higher

For network communication (one or more of the following):

- Digital TCP/IP services for OpenVMS
- TCPware for OpenVMS
- Multinet for OpenVMS
- DECnet Phase IV
- DECnet Plus

## Privilege Requirements

To ensure that OSM runs correctly, the process that starts both the monitor and transport process must have:

- Access to the directory containing all OSM data files and executable images
- Access to the UIC group 1 logical name table
- ACNT and DETACH privileges

The process that stops either the OSM monitor process or the OSM transport process should also have the WORLD and SECURITY privilege. The SECURITY privilege is needed to verify that the security audit listener mailbox is properly disabled by OSM. This prevents the AUDIT_SERVER process from hanging while attempting to write a message to that mailbox.

The OSM monitor process must have the CMKRNL, DIAGNOSE, LOG_IO, NETMBX, PRMMBX, SECURITY, SYSLCK and SYSNAM privileges. The OSM transport process requires the SYSLCK privilege.

Some OSM commands and automated fix commands, such as the OSM command SET TIME, may require additional privileges.

**Note:** OSM will automatically create the transport or monitor process with the correct privileges if the account creating either of these processes is authorized to enable these privileges; either the account has SETPRV or the account has the authorized privileges needed to run the OSM processes. OSM will turn off the SHARE and BYPASS privileges for both the monitor and transport processes during startup. The OSM images are installed with the correct privileges by the OSM_STARTUP.COM command procedure.

If OSM cannot enable any of the privileges, it will continue processing. Attempts to perform some functions may fail if the monitor or transport does not have the required privileges. If a function fails due to lack of privileges, a message will be displayed in the processes log file (OSM_TRANSPORT.LOG or OSM.LOG) stating that not all privileges were enabled.

# Chapter 2
# OSM Operation

## Overview

This chapter discusses the general operation of OSM. Overall OSM operations are controlled by user-definable parameters that are placed in one of two parameter files. The OSM monitor and transport processes perform the majority of the functionality in OSM.

The OSM monitor process, named OSM, will continuously monitor and record all exception events. An exception event is one of the monitored OSM events, or a user-defined event that has exceeded a user-defined threshold. For example, if the user has specified that a disk should not be more than 80% full, OSM will automatically notify the destination(s) specified in the parameter file when that disk has exceeded that threshold (the exception event). All of the collected exception event information will be stored in a file with a file extension of ".ACTIVE". For more information on the events that OSM can monitor, refer to *Chapter 1: Introduction to OSM* in this manual.

Once OSM has detected an exception event, it can do one or more of the following:

- **Notify** OSM can notify (report to) the specified personnel of the exception event. This can be accomplished by sending E-mail locally, over the network or Internet, or by uploading the information to a central facility over a network.

  When exception event notification occurs is dependent on whether the event is critical or non-critical. If the event is critical, notification occurs immediately. All non-critical exception events are stored and sent once a night.

  The monitor process initiates exception event notification. When it determines that exception event information needs to be sent, it will rename the file it has stored event information in from a file type of .ACTIVE to a file type of .SEND. The transport process then wakes up to process the .SEND files. It does this by sending the information contained in the file(s) to the location specified in the OSM parameter file(s).

- **Page** OSM can automatically page a specified party to inform them of critical events. OSM supports multiple paging attempts on a single pager, and paging a backup pager if the initial pager request is not responded to within a specified period of time.

  The monitor process usually initiates this. If an exception event is detected that has a page associated with it, the monitor process will rename the current .ACTIVE file to a .SEND file. This file will have the information about the exception event as well as an indication that a page must be invoked. The transport process then wakes up, finds the .SEND file, processes the information, and invokes the page.

- **Fix** OSM can call a user-specified command, command procedure or batch file to fix either critical or non-critical exception events. The monitor process performs this. When an exception event occurs, the monitor process looks at the fix parameter associated with that event. If the fix parameter value has been specified, OSM will execute the fix command or command procedure in the context of a subprocess. Fix parameter flags allow the user to specify that the fix be performed while the monitor process continues processing, or can be set so that the monitor process suspends execution until the fix is complete.

Detection of the events and execution of the fix command is performed by the monitor process. The OSM transport process performs actual paging and event notification. The monitor process must run on each node to be monitored by OSM. In a VMScluster environment only one transport process is needed to perform paging and event notification for all nodes in the cluster. For a standalone node, both the monitor and transport process must be running on that node.

Both processes are controlled by parameters that are loaded from the parameter file(s) during process startup. For more information on the OSM parameter files, see *Chapter 3: OSM Parameter Files and Formats* of this manual. A list of all OSM files, including the parameter files, and their purposes are included in *Appendix B* of this manual.

# OSM_CONTROL Image

The OSM_CONTROL image is the interactive interface for OSM. There are three versions of this image. OSM_CONTROL allows you to:

- Start, stop and wake the OSM processes
- Show the status of OSM
- Edit OSM parameter files
- Display the monitor or transport process parameters currently in use
- Cause OSM to immediately process any collected event information
- Show and clear paging requests and history
- Display or synchronize time from a remote node running OSM
- View the temperature reported by a USB temperature sensor
- Initiate synchronization of specific accounts in the SYSUAF
- Summarize previously collected exception event data
- Get help on OSM commands

## OSM_CONTROL

OSM_CONTROL is activated through the use of a foreign command that is created as documented in *Chapter 15: OSM Installation* in this manual. The OSM_CONTROL image can either be run interactively or by issuing a single OSM command at the DCL Prompt. To issue commands interactively, simply type OSM at the DCL prompt. The utility will identify itself and generate the OSM> prompt; for example:

```
$ osm
Welcome to Online System Manager V4.2
Copyright @1992-2024 The PARSEC Group

OSM is licensed for both software and hardware

OSM>
```

At the OSM prompt you can issue any valid OSM command, qualifier, and parameter, terminating the command with a carriage return. After each command execution, the OSM_CONTROL image will return to the OSM> prompt for another command. When you are done executing OSM commands issue the EXIT command or press the control-Z key to cause the OSM_CONTROL image to return to the DCL prompt.

While in OSM, normal DCL style command line editing is available. OSM has a 100-command line recall buffer and supports DCL style command line editing. This allows you to use the arrow keys to reposition the cursor and make corrections in the command just typed. By pressing the up or down arrow key, you may recall any of the previous 100 command lines entered, then edit that command if desired, and re-execute the command by pressing the carriage return key.

To issue a single OSM command from the DCL prompt, type the foreign command followed by the desired OSM command string. For example:

```
$ osm show status
Online System Manager V4.2

OSM Transport:  not running
OSM Monitor:    V4.2           20361B16 on node CLASS8 username WILLIAMS
OSM Control:                   2036A446 on node CLASS8 username WILLIAMS
```

OSM is the foreign command, and SHOW STATUS is the OSM utility sub-command. After execution of the OSM sub-command, the user is returned to the DCL prompt.

# The OSM Help Command

## HELP

This command will allow you to get help on any of the OSM control commands.

**Format**

**HELP [topic [subtopic ...]]**

**Parameter**

**topic**

Specifies the topic for which help is desired. If this parameter is not specified, then a list of topics for which help is available will be displayed. If a topic is specified with subtopics, then help is displayed for that subtopic.

# Chapter 3
# OSM Parameter Files and Formats

## Overview

All aspects of OSM are controlled by user-definable parameters to allow you to tailor OSM for your environment. To have OSM recognize your parameter values, they must be placed in a parameter file along with their associated value.

OSM provides two parameter files. The purpose for having multiple files is for a cluster configuration, where you have multiple nodes referencing the same OSM directory. Having one generic file for the whole cluster allows you to define default parameters for the cluster. Having a node-specific file allows you to override those defaults for a particular system.

All parameter files reside in the directory pointed to by the logical name OSM$. Executing the OSM command SHOW PARAMETERS allows the user to see the names of each parameter file that is used. The output from this command includes the name of each parameter file and the parameters in each file when OSM was last started. The name of the common parameter file is OSM_CTLFILE.DAT. The name of the node-specific parameter file is OSM_CTLFILE_nodename.DAT where nodename is the name of the node running the OSM application. If the node name has not been defined via the SCSNODE parameter in SYSGEN, the DECnet node name will be used. If a node name has not been defined, the node-specific file name will be OSM_CTLFILE_.DAT. If OSM finds a parameter specified in both the node-specific parameter file and the cluster-wide parameter file, it uses the value specified in the node-specific parameter file.

### Parameter file format

The first five lines in the cluster-common parameter file are customer name, three lines of address and contact name. The remaining lines in the cluster-common parameter file, and all lines in the node-specific parameter file(s), are keyword-controlled parameters. Any line may include trailing blanks or tabs and may also include comments. Comments are included by inserting text after the "!" character. A line may also include only comments by beginning the line with the "!" character. Keyword controlled parameters are separated from the value by an equal sign (=). Refer to the example at the end of this chapter to see a sample cluster-common parameter file.

**Note:** The customer’s name, address and contact information must be in the cluster-common parameter file, not the node-specific parameter file. All other parameters may reside in either parameter file.

This manual has a chapter for each portion of OSM functionality. In those chapters, there is a section on the command and a section on the parameters that affect that portion of OSM's

functionality. Most of the parameters have default values and if so, the default values are displayed with the parameter. Any parameter value specified in one of the parameter files will be used instead of the default value.

# OSM Parameter Formats

Each OSM component is controlled by parameters to allow you to customize it for maximum use in your environment. Each of these components' capabilities and parameter values is discussed in detail in later chapters. The following OSM components have their own set of parameter values:

- Identification
- Controlling OSM Processes
- Monitoring for Exception Events
- User-Defined Event Monitoring
- Network Event Monitoring
- Exception Event Reporting
- Paging Support
- Automated Fixes
- SYSUAF Synchronization
- Monthly Summary Reporting

The basic types of parameters are boolean, numeric, string, keyword, time and severity level/threshold parameters. If there are multiple values for a parameter, the values are separated by a comma. Some parameters combine different types of values in one parameter. An example of this is the PAGER parameter that has a numeric value and a string value separated by a comma.

## Numeric parameters

Numeric value parameters will use the specified number in controlling how the OSM monitor process deals with the parameters. These parameters may specify an amount of time in seconds or days, a count, a severity level, or just a raw number. An example of a numeric threshold would be the CRITICAL parameter that specifies which severity levels are considered critical.

## String parameters

String value parameters use the specified string in controlling how OSM deals with the parameters. Some examples of string values would be commands to be executed, customer information and file names. An example of a string parameter is TEXT_PREFIX. This specifies a string of text to be included in formatted messages such as E-mail and printer output.

## Boolean parameters

Boolean parameters either enable or disable collection of the specified data or enable or disable some other function in OSM. Valid values are an ASCII "Y" or "y" that enables collection of

the specified data, or an "N" or "n" that will disable collection of the specified data. The OPCOM parameter is an example of a Boolean parameter. The default is OPCOM=Y which allows the OSM transport process to send an OPCOM message each time it starts processing files. This may be changed to OPCOM=N which will not allow the transport process to send this message.

### Keyword parameters

Keyword parameters are a special case of string parameters. They will only allow the keywords that are valid for that parameter. An example of a keyword parameter value would be the OUTPUT parameter. This parameter accepts keywords specifying where to send OSM collected information.

### Time parameters

Time value parameters specify an interval of time. Depending on which parameter is being specified, the time entered may be either in hh:mm to specify hours and minutes or in the OpenVMS delta time format of dd hh:mm:ss.cc to specify the number of days, hours, minutes, seconds, and hundredths of seconds.

An example of a time value-based parameter is the PAGER_RETRY parameter. This parameter specifies how frequently OSM should retry a pager request that has not been cleared.

### Severity level/threshold parameters

Severity level parameters are used to define the severity level for certain events. Events with severity levels that are considered normal will be handled differently than events with high severity levels. For severity, the minimum value is 0 and the maximum value is 10. Any value greater than 10 is set to 10 while any value less than 0 is set to 0. Setting the severity of an exception event to 0 will disable the reporting of that exception event. Setting the severity to 1, flags that event as low severity while setting the severity to 10 flags that event as a high severity event, which is critical to the operation of the system.

Threshold parameters include the severity level for an exception event and the threshold that is considered normal for that event. The OSM monitor process uses threshold parameters to determine what an exception event is. For those parameters that are percentages, any value less than 0 is set to 0 and any value greater than 100 is set to 100.

An example of a threshold parameter is DISK. This parameter specifies the severity level associated with any disk exception event, along with what minimum amount of free disk space that should be allowed before reporting a disk as an exception.

## OSM Parameter File Commands

The following OSM commands are used to modify and deal with OSM parameter files.

### EDIT PARAMETERS

This command allows the user to modify the parameters that control OSM. There are at least two parameter files, the node specific parameter file and the common parameter file.

This command invokes an editor for modifying the OSM parameter file. This command may edit the cluster common parameter file, or a node specific parameter file for a node in the cluster. The editor invoked will be determined in the following order:

1) The OSM Command SET EDITOR
2) The OSM parameter EDITOR
3) The logical name OSM$EDIT
4) The default editor of TPU

**Format**

**$ OSM EDIT PARAMETERS**

**Qualifiers**

**/NODE[=nodename]**

Specifies which node's parameter file is to be edited. If you specify this command without the /NODE qualifier, the cluster-common parameter file will be edited. If specified with the /NODE qualifier, but no node name, the node-specific parameter file for the current node will be edited. If specified with the /NODE qualifier and a node name, the node-specific parameter file for that node will be edited.

## EXIT

This command exits the OSM_CONTROL application. This also can be performed by pressing control/Z at the command prompt.

**Format**

**EXIT**

## READ PARAMETERS

Each of the OSM images reads the current parameter files when it starts up. This command causes the OSM_CONTROL program to reread the current parameter files and use the parameters in them. This may be useful to test any changes made to either the cluster-common or node-specific parameter file. It is also necessary to issue this command after making changes to any of the contact information in the parameter file before issuing the SEND CONTACT_INFORMATION command or when changing the network ID before setting the network password.

**Format**

**READ PARAMETERS**

**Note:** The READ PARAMETERS command does not cause the new parameters to be used by either the monitor or transport process. Any processes affected by the modified parameters must be stopped and restarted before the parameter change will become effective. See the STOP MONITOR, START MONITOR, STOP TRANSPORT and START TRANSPORT commands in *Chapter 5: Controlling OSM Processes* for more information.

## SEND ALL

This command causes OSM to send all messages that have been collected by OSM but have not yet been sent. These are messages that would normally be sent that night.

**Format**

**SEND ALL**

**Qualifiers**

**/NODE=nodename**

The /NODE qualifier allows the user to specify which node's messages will be sent. The command will be executed on the selected node, but that node may be able to send messages for another node that shares the same directory. Standard OpenVMS wildcards of * and % are allowed in the node name. The current node is the default.

## SEND REQUEST

This command allows the user to send a text message to the OSM monitor process. This message may be sent instead of making a phone call to the recipient of OSM messages. This command will place the sender into the selected editor and allow him to enter text. The editor used is the same as in the OSM command EDIT PARAMETERS.

**Format**

**SEND REQUEST**

**Qualifiers**

**/[NO]ALL**

Causes all previously collected, but not yet sent, information to be included with the new data being sent. The default is /NOALL.

**/[NO]NOW**

Causes the information to be sent immediately and is the default. /NONOW causes the information to be included in the binary event file. This data is sent the next time that the transport process is required to send the binary event file.

## SEND TUNING_INFORMATION

This command sends basic tuning information as described for the CAPACITY parameter in *Chapter 6: Monitoring for Exception Events*. The data is sent to the transport process for immediate processing.

**Format**

**SEND TUNING_INFORMATION**

## SET EDITOR

This command sets the editor to be used by OSM for the EDIT and SEND REQUEST commands listed above without changing the editor specified in the parameter file for other users. This may be any editor that is supported by OpenVMS mail, or it can be specified as @file-spec to invoke a DCL command procedure for editing a file. This includes the EDT, TPU and TECO editors that are provided with OpenVMS, and the LSE editor which is a layered product available from HP. If you exit and reenter the OSM_CONTROL image, the editor used reverts back to the editor defined by the OSM$EDIT logical name or the EDITOR parameter. To see the editor to be used by OSM, issue the OSM command SHOW EDITOR.

If this command is not executed after entering the OSM command line interface, the editor to be invoked when editing anything is determined first by the OSM parameter EDITOR, second by the OSM$EDIT logical name and last by the default editor TPU.

**Format**

**SET EDITOR editor-name**

## SHOW EDITOR

This command will display the editor to be used when performing any edit function within OSM. The editor can be set with the OSM command SET EDITOR, the OSM parameter EDITOR or the logical name OSM$EDIT.

**Format**

**SHOW EDITOR**

## SHOW PARAMETERS

This command displays the last parameters loaded by either the OSM monitor or transport process. This will not list all parameters used, but only those specified in one of the parameter files.

If the output is more than one screen of information, OSM will pause at the end of each screen. Pressing the RETURN key will cause the next screen of information to be displayed. Pressing the control/Z key will cause the display of any more information from this command.

**Format**

**SHOW PARAMETERS**

**Qualifiers**

**/ALL**

This qualifier causes the command to display all of the parameters in use by the current process; not just those included in a parameter file. This allows viewing the effective parameters, including those defaulted.

**/MONITOR**

This qualifier causes the command to display those parameters used by the OSM monitor processes when it was last started. The default is /TRANSPORT.

**/TRANSPORT**

This causes the command to display those parameters used by the OSM transport process when it was last started. This is the default.

**/NODE[=nodename]**

This qualifier is only valid for viewing OSM monitor process parameters. It can be used to display the monitor process parameters for the specified node in the VMScluster. If no node name is specified, the qualifier defaults to the node on which the command was executed. The /NODE and the /TRANSPORT qualifiers are mutually exclusive.

**/PAGE_SIZE=number-of-lines**

This specifies the number of lines to be displayed per page before prompting the user to press ENTER for the next page. The default is the page length for the current terminal.

# Parameter File Parameter

## EXIT

This parameter causes OSM to exit the current parameter file. This allows some parameters to be included in the file but not currently used. When OSM reads the parameter files it starts at the beginning of the file and translates each parameter in order. It stops when it reaches the end of the file or the EXIT parameter. This allows you to try setting a parameter and if you do not like the results you can leave the parameter in the file but insert the EXIT parameter before it. The purpose for this would be to keep a chronological history of parameter settings that you did not like, preventing you from forgetting what you had tried and trying it again.

# Example Control File

The following is an example cluster-common control file. Although it is relatively simplistic, it illustrates how easy it is to control OSM's behavior. Keep in mind that any parameter that is not specified in one of the parameter files will revert to the defaults shipped with OSM. The defaults for each of the parameters are documented with the parameters in their respective chapters in this manual and in *Appendix A*. A node-specific control file has the same format but will omit the first five lines of contact information.

```
My Company                  ! Company name
1 N Main St.                ! and address
Local city, CO  80111

John Adams                  ! Contact for support
Phone=303-555-1212          ! Voice phone
Fax=303-555-1210            ! FAX phone
account=9999                !
editor=TPU
output=mail                 ! Send files to mail distribution
mail=jadams                 ! Send mail to username jadams
pager_retry=3,0 0:0:30      ! retry pager 3 times, each 30 seconds apart
audit_breakin_p=97
```

# Chapter 4
# OSM Identification Parameters

## Overview

This chapter describes the OSM identification parameters. These parameters are useful if OSM is running on and monitoring several remote sites and sending all exception event data to a central location. In this type of environment, it is useful to have information such as remote site contact name, phone number, FAX number as well as other pieces of information. That way, if the central management team received exception information from that site and they either needed more information or someone local to do something, they will know who to contact and how.

The identification information for each site must be set up on that site. This is accomplished by defining parameters in one of the OSM parameter files and the first five lines of information in the cluster-common parameter file.

To define a contact parameter, edit the desired parameter file and type in the parameter name along with its value, for example: FAX=303-515-1515. The first five lines of the common parameter file are filled in by the OSM installation procedure based on what the user entered in response to the contact questions. Editing the common parameter file allows the user to change this information. It is a good idea to keep this information current. For more information on parameter files, refer to *Chapter 3: OSM Parameter Files and Formats* in this manual.

If you are running OSM on a standalone system or in a VMScluster environment and are reporting only to a node in the cluster, you will probably not use this portion of OSM. However, the first five lines of the cluster-common parameter file must still contain the required identification information.

All or part of this information can be sent to you in mail when a user executes the OSM command SEND CONTACT_INFORMATION. Some of this information may be included in each mail message OSM sends by including the keyword CONTACT in the parameter list for the OSM parameter DETAIL (DETAIL=CONTACT).

# Identification Commands

## SEND CONTACT_INFORMATION

This command sends the current contact information to the location specified to receive exception event information.

**Format**

**SEND CONTACT_INFORMATION**

**Qualifiers**

**/[NO]ALL**

Causes all previously collected, but not yet sent, information to be included with the new data being sent. The default is /NOALL.

**/[NO]NOW**

Causes the information to be sent immediately and is the default. /NONOW causes the information to be included in the binary event file. This data is sent the next time that the transport process is required to send the binary event file.

# Contact Parameters

The following parameters, along with the first five lines of the common parameter file, make up the contact information.

## Organization name

The organization name is not a keyword-based parameter, but is the first line of the common parameter file. This is included in mail messages generated by the OSM command SEND CONTACT_INFORMATION. It is also included in exception event mail messages if the OSM parameter DETAIL has the keyword CONTACT. Assign this parameter the value of the name of the organization that owns the system on which this parameter is defined.

## Address

The address is not a keyword-based parameter, but is the second, third and fourth lines of the common parameter file. They are included in mail messages generated by the OSM command SEND CONTACT_INFORMATION. Assign these parameters the value of the address of the organization that owns the system on which this parameter is defined. If the address has fewer than three lines, leave the remaining lines blank.

## Contact Name

The "contact name" is not a keyword base parameter, but is the fifth line of the common parameter file. This is included in mail messages generated by the OSM command SEND CONTACT_INFORMATION. It is also included in exception event mail messages if the OSM parameter DETAIL has the keyword CONTACT. Assign this parameter the value of the name of the person who is locally responsible for maintaining the system on which this parameter is defined.

## ACCOUNT=nnn

This parameter specifies the customer access number for the current node. There is not a default value for this parameter. This is the only keyword-controlled parameter that is required. The value of this parameter is included in every mail message generated by OSM. This should be specified as a number in the range of 1 to 10000. The installation procedure includes this parameter in the node-specific parameter file. If the account number is not specified or is specified as 0, all OSM applications will abort with a status of SS$_BADPARAM.

## FAX=nnn-nnn-nnnn

This is the FAX number for the person responsible for this system. This parameter is included in the cluster-common parameter file during the installation of OSM if a FAX number is provided. The value of this parameter is included in mail messages generated by the OSM command SEND CONTACT_INFORMATION

### MODEM=nnn-nnn-nnnn

This is the modem number for this system. This is included in mail messages generated by the OSM command SEND CONTACT_INFORMATION. This parameter is included in the cluster-common parameter file during the installation of OSM if a value is specified.

### PHONE=nnn-nnn-nnnn

This is the phone number for the person responsible for this system. This is included in mail messages generated by the OSM command SEND CONTACT_INFORMATION. It is also included in exception event mail messages if the OSM parameter DETAIL has the keyword CONTACT. This parameter is included in the cluster-common parameter file during the installation of OSM if a value is specified.

### SITE=

This parameter affects the node name used by OSM to identify the system. This will be used in naming the binary data files sent by OSM and in mail messages when formatting data for this node.

On OpenVMS, OSM tries to get the node name first from the SYSGEN parameter SCSNODE, and next from the DECnet node name. This parameter is only used if the SYSGEN parameter SCSNODE is not defined and DECnet is not set up and running. If the SITE parameter is not defined, and is needed, a blank node name will be used. This parameter has no default value.

### TEXT=

This parameter allows you to add a single line of text to be used for whatever purpose you desire. An example for using this would be to identify a location within the building of a particular system such as, TEXT=Computer in Cubicle A3

### TEXT_ADD=

This parameter allows you to add additional lines of text to be used for whatever purpose you desire. An example for using this would be to identify a secondary person responsible for a particular system. This parameter only applies if you have the TEXT = parameter already.

# Chapter 5
# Controlling OSM Processes

## Overview

This chapter discusses how to start, stop, and wake any of the OSM processes. Starting the OSM monitor process is required before OSM can monitor the system. Stopping the OSM processes will terminate OSM actions on that system or cluster.

### Starting and Stopping OSM Processes

Starting the OSM monitor process is normally performed when the system boots. Once the OSM monitor process has been started, it will start the transport process if enabled by the AUTOSTART parameter.

Once the OSM startup procedure has executed OSM processes can be stopped and restarted by executing the OSM stop and start commands, which are documented later in this chapter. During system shutdown, the operating system automatically stops all processes, including OSM.

The OSM monitor, transport and server processes should be stopped and restarted when:

- **There have been parameter changes.**
  All OSM parameters are static. If you modify a parameter, you must stop and re-start the affected OSM processes before the new parameter value takes effect.
- **When changing the network communication password for OSM.**
  OSM secures its network communication with a password. That password is loaded when the OSM monitor process is started. If the password is changed with the OSM command SET PASSWORD, the monitor process must be restarted.
- **When changing the logical name which defines the parameters for a disk.**
  OSM uses a set of logical names to define the thresholds, severity, pager notification string, and automated fix commands on a per disk basis. These logical names are translated when the monitor process first detects the existence of the specified disk. The logical names used are OSM_DISK_* and are documented later in this chapter.

- **Performing a backup.**
  Performing a backup using an account that has one or more of the file access privileges enabled can generate a large number of security alarms if those alarms are enabled. If you do not want OSM to report on those alarms, you can stop the monitor process for that node or disable those alarms during the backup.
- **Restarting the security audit server.**
  OSM notifies the security audit server to provide it a copy of security event messages. If the audit server is restarted, then OSM needs to be restarted to be notified of future security events.
- **Performing an OSM upgrade.**
  The OSM upgrade procedure automatically stops and restarts the OSM processes.
- **Performing an OpenVMS upgrade.**
  Depending on what OpenVMS version you are upgrading from and what OpenVMS version you are upgrading to, OSM may need to be re-linked as well. OSM must be stopped and restarted before the new executable image will be run.

There are specific privilege requirements for the OSM monitor and transport processes and for the process that starts and stops OSM. These are listed in detail in *Chapter 1: Introduction to OSM*.

## Purging OSM Processes Working Sets

Built into both the transport and monitor process is the ability to purge their working sets before hibernating. This is controlled by the OSM parameter PURGE_WORKING_SET. By default this has a value of N, indicating that it is disabled.

Unless you are short on memory, automatic working set purge should be disabled since both the transport and monitor processes would have to generate additional page faults to populate their working set whenever they wake up.

If you are short on memory, there is no reason for either of these processes to have a large working set while hibernating. So in that case, you can save memory usage by setting the OSM parameter PURGE_WORKING_SET to a value of Y.

## Waking the OSM Transport Process

Normally the OSM transport process is sleeping, waiting to execute. It periodically wakes up; based on the OSM parameter TRANSPORT_WAKE that defaults to a 5-minute interval, to determine if there are any event files to be processed. The OSM monitor process wakes up the transport process whenever it creates a file with data to be processed. Processing the binary event files can lead to reports on exception events or pages or both.

You may manually wake the transport process if there are binary event files that need to be processed before the next scheduled transport wakeup interval. These files may have been transferred to a system from another system via the network. This is accomplished by the OSM command WAKE TRANSPORT, discussed later in this chapter.

## Automatic Startup During Boot

OSM may be restarted each time the operating system is booted. Add the command @SYS$MANAGER:OSM_STARTUP to the site specific startup procedures, named SYS$MANAGER:SYSTARTUP_VMS.COM. This procedure may also be added to the SYSMAN startup database instead of the site-specific startup procedures. To stop having OSM restarted each time OpenVMS is booted, remove the command procedure from the site-specific startup procedures and the SYSMAN startup database.

## Automatic Failover

In VMScluster environments, there is only one transport process running on the cluster. It is responsible for processing all cluster member's monitor process's binary event files. If the transport process terminates, or the node on which the transport process is running goes away for any reason, the binary event files will not be processed. This would mean that pages would not be invoked and exception events would not be reported. This could be very bad if it happened at night or some other time when no one was around to notice that the transport process was not running and restart it.

To eliminate this potential problem, OSM allows for the transport process to automatically fail over to another node in the cluster. Implementing automatic failover is the responsibility of the OSM monitor process.

The monitor process periodically checks for the existence of the transport process. If the transport process is not running, the monitor process can automatically start the transport process. This is controlled by the OSM parameter AUTOSTART and the OSM commands ENABLE AUTOSTART/TRANSPORT and DISABLE AUTOSTART/TRANSPORT. The AUTOSTART parameter affects the monitor on the node on which the parameter is defined. The enable and disable autostart commands affect the entire cluster. The best way to describe the functionality is through scenarios.

In the first scenario, autostart has not been disabled via the OSM command and all nodes in the cluster have their AUTOSTART parameter set to a value of 300,TRANSPORT. In this example, each individual monitor process in the cluster checks for the existence of the transport process every 300 seconds. The first monitor process to detect that the transport process does not exist in the cluster creates it on that node. If, in this scenario, the DISABLE AUTOSTART/TRANSPORT command is executed on any node in the cluster, none of the monitor processes will attempt to restart the transport process until autostart is re-enabled with the OSM command ENABLE AUTOSTART/TRANSPORT.

In the second scenario, autostart has not been disabled with the OSM command, and all nodes in the cluster except one, have their AUTOSTART parameter set to a value of 300,TRANSPORT. The other node has its autostart parameter set to 0, which disables autostart on that node. The nodes that have their AUTOSTART parameter set to 300 check once every 300 seconds for the existence of the transport process. If it does not exist, it will be started on the node that first detects it was not running. The monitor process that is running on the node with the AUTOSTART parameter disabled will not check to see if the transport process exists; therefore, it will never start the transport process on this node.

You disable autostart via the AUTOSTART parameter for nodes that you do not want the transport process to run on.

You may temporarily disable autostart with the OSM command DISABLE AUTOSTART/TRANSPORT to prevent the transport process from re-starting if you stopped the transport process for some reason. If you did not disable autostart, the next time a monitor process that had their AUTOSTART parameter set to a value other than 0 checks for the existence of the transport process, it restarts it on that node. You then re-enable autostart with the OSM command ENABLE AUTOSTART/TRANSPORT when you want OSM to resume processing .SEND files.

## Automatic Shutdown During System Shutdown

OSM may be shut down each time the operating system is properly shut down. Add the command @SYS$MANAGER:OSM_SHUTDOWN to the site specific shutdown procedures, named SYS$MANAGER:SYSHUTDWN.COM.

## OSM Logical Names

OSM for OpenVMS uses several logical names to aid the system manager in setting up and controlling OSM. Users may enhance the logical names created on OpenVMS by editing the OSM_LOGICALS.COM file in the OSM$ directory.

By default, these logical names are defined in the system logical name table. If the parameter LNM_TABLE has been set in OSM_CTLFILE.DAT, then that parameter determines in which logical name table they are defined and will be used when looking up the logical names. A few of the logical names are always be stored in the system logical name table.

The following list describes the logical names associated with OSM and their uses.

**OSM$**

This is the main logical name used by OSM on OpenVMS. It points to the device and directory that contains all OSM files including: OSM executables, OSM parameter files, OSM .SEND and .ACTIVE binary event files, and any OSM command procedures. This logical name is always defined /SYSTEM.

**OSM$EDIT**

This logical name determines the editor to be used by the OSM command EDIT. The default editor can also be set with the OSM command SET EDITOR, or the OSM parameter EDITOR. If the default editor has not been defined by any of these mechanisms, the default will be TPU.

The OSM parameter EDITOR will override the setting of the OSM$EDIT logical name. The SET EDITOR command will override an editor defined by either the EDITOR parameter or the OSM$EDIT logical name.

The reason for this hierarchy is to allow the editor to be selected on a per user basis. The best way of accomplishing this is by defining the logical name OSM$EDIT system-wide for all users but having specific users define the logical name in their process or job logical name table. For this to work, the EDITOR parameter must not be in the OSM_CTLFILE.DAT file. To determine which editor is being used, issue the OSM command SHOW EDITOR.

**OSM$LOCAL**

In a VMScluster environment, this logical name points to an OSM$ subdirectory. In a non-clustered environment, it points to the OSM$ directory. It is the directory that user-defined event monitoring files must be placed in before OSM can recognize and process them. It will also hold some of the various temporary files created and deleted by OSM. For more information reference *Chapter 7: User-defined Event Monitoring* in this manual. This logical name is always defined /SYSTEM.

**OSM$LOG**

This logical name may be defined before the startup of OSM to reference the directory to contain the log files from the monitor and transport processes. If not defined, then this will be defined to point to the OSM$ directory. This logical name is always defined /SYSTEM.

**OSM_***

The OSM_STARTUP procedure creates logical names for any OSM compatible products that have been installed on the system. The list of products installed is determined by the subdirectories OSM_*.DIR that are found in the main OSM directory. For example, an OSM demonstration package may be installed in the [.OSM_DEMO] subdirectory to the main OSM directory. The OSM startup procedure would create the logical name OSM_DEMO. A summary utility which works with OSM might be installed in the [.OSM_SUMM] directory and would have the logical name OSM_SUMM associated with it.

If a subdirectory pointed to by an OSM_* logical name, created by the OSM_STARTUP procedure, contains a startup procedure for the OSM supported product in that directory, it will also be executed before the OSM images are started. The name of the startup procedure will be the same as the name of the directory followed by _STARTUP.COM. For example, for the OSM demonstration package, the startup procedure would be named OSM_DEMO:OSM_DEMO_STARTUP.COM.

**OSM_AUDIT_LISTENER**

This is the logical name of the mailbox to which the AUDIT_SERVER process sends security messages to OSM. Setting the parameter AUDIT=N will prevent OSM from creating this mailbox and logical name. This logical name is always defined /SYSTEM.

**OSM_COMMAND**

This logical name references an OpenVMS mailbox used for communication between the OSM Control utility and the OSM monitor process. This logical name is always defined /SYSTEM.

**OSM_CONTROL**

This logical name references the correct image to be run for the current architecture and version of OpenVMS. It is referenced by the OSM global symbol. This logical name is always defined /SYSTEM.

**OSM_DISK_***

These names can be used to override the default way of monitoring disks on a disk-by-disk basis. The asterisk (*) would be the name of the disk for which you wish to override the monitoring parameters. The default monitoring parameters for all disks are determined by the OSM parameters DISK, DISK_C, DISK_P, DISK_F and DISK_CF. The format for this logical name is:

```
$ OSM_DISK_xxxx p1,p2,p3,p4,p5,p6,p7,p8,p9,p10
```

The following table describes the parameters:

| Parameter | Value |
|---|---|
| xxxx | Name of the **physical** disk device that this logical name affects. |
| p1 | The percentage of disk to be used for notification. |
| P2 | The percentage of disk to be used for critical notification. |
| p3 | The number of I/Os per second for notification. |
| p4 | The number of I/Os per second for critical notification. |
| p5 | The non-critical severity level. |
| p6 | The critical severity level. |
| p7 | The pager notification string used to generate a to page. |
| p8 | The flags for the subprocess that executes any fix commands. |
| p9 | The fix command string. |
| p10 | The critical fix command string. |

The following logical name would allow you to override the way OSM monitors, reports, pages and fixes exception events detected for the disk device $1$dka400:

```
$ define/system/executive_mode OSM_DISK_$1$DKA400 -
_$ 95,99,40,300,2,7,smith@parsec.com,0,"@osm$:disk_fix", -
_$ "@osm$:disk_critical_fix"
```

In this example, a non-critical disk usage event is generated when the disk exceeds 95% full. A critical disk usage event is generated when disk space exceeds 99% full. A non-critical event will be generated when there are more than 40 I/Os per second, and a critical event will be generated when there are more than 300 I/Os per second. The non-critical severity level associated with the events is 2, and the critical severity level is 7. The pager notification string is smith@parsec.com. There are no flags for the fix command subprocess. The command string to execute when a non-critical exception event is generated is @OSM$:DISK_FIX and the command string to execute when a critical exception event is detected is @OSM$:DISK_CRITICAL_FIX.COM.

If a portion of the logical name is not to be defined, then replace it with an empty quoted string (" "). This would be necessary if the pager notification string is to be omitted, but an automated fix command string is specified. If an automated fix command string includes blanks or the "@" or "!" characters, it must be enclosed with quotes. The automated fix command string executed for disk events will have the name of the disk appended to the end of the command. If that is not needed, then end the command string with the comment character (!). For additional information about the pager notification string, reference *Chapter 10, OSM Paging Support*. For additional information about the automated fix command strings, reference *Chapter 11, Automated Fixes*. Once one of these logical names has been defined, or redefined, the OSM monitor process must be stopped and restarted.

**OSM_ERRFMT**

This logical name points to the mailbox to which the ERRFMT process on OpenVMS sends messages to be read by OSM. Setting the parameter ERRFMT=N will prevent OSM from creating this mailbox. This logical name is always defined /SYSTEM.

**OSM_LOCAL**

It is the directory that user-defined event monitoring files must be placed in before OSM can recognize and process them. It will also hold some of the various temporary files created and deleted by OSM. For more information reference *Chapter 7: User-defined Event Monitoring* in this manual.

**OSM_SYSUAF_***

This set of logical names is used by OSM to assist the system manager in maintaining synchronization of system authorization files in a network. Refer to *Chapter 12: SYSUAF Synchronization* for a complete description of this feature and the associated logical names.

**OSM_VER**

This name is assigned to the version of OSM that was last started. For example, its translation will be 0420 for V4.2, for V2.4-1 it will translate to 0241.

**OSM_VMS_VER**

This logical name is created by the OSM_STARTUP procedure, as a logical name search list. The first translation of the logical name equates to the version of OpenVMS. The second translation of the logical name equates to the type of architecture on which it is executing; VAX, Alpha or Integrity server. An Alpha system is designated as AXP in this logical name. An Integrity server is designated as I64 in the logical name. The third translation of the logical name equates to the last portion of the name of OSM executable images to be run.

**OSM_LOGICALS.TEMPLATE**

This file is used to provide you with a common place to put all OSM related logical names that are user-definable. It also includes comments describing the logical names that may be modified. To use this file, copy the template file to OSM_LOGICALS.COM. Then add all of your name definitions to it.

# Multi-Environment Example

This section describes setting up OSM to run on a set of networked computers running OpenVMS VAX, OpenVMS Alpha and OpenVMS Integrity servers. The OpenVMS systems include both clustered and non-clustered nodes. The setup described here includes having OSM data reported for all nodes, having OSM monitor network accessibility of all nodes expected to be running and synchronizing time on all nodes to a common source. Once this example is understood, adapt it to satisfy your situation.

## Node Descriptions

There are three stand-alone server nodes named VPROD1, VPROD2 and VPROD3. These systems are relied on most heavily by the organization and are always expected to be running. If one of these systems goes down, every effort is made to fix the problems and get the systems back into production.

There are four systems named VTEST1, VTEST2, VTEST3 and VTEST4. These systems are used for experimental testing and troubleshooting of the applications. These systems may be rebooted as needed and may be left down for extended periods of time.

There are three OpenVMS Alpha and one OpenVMS VAX workstations. These systems are named VMSWS1, VMSWS2, VMSWS3 and VMSWS4. These systems are used for software development and other interactive applications and are expected to be kept running continuously. Each of these systems has DECnet and Digital TCP/IP Services for OpenVMS installed and running for file transfers and network communication.

There are two OpenVMS Integrity and one OpenVMS Alpha systems running in a cluster. They are named VMSCL1, VMSCL2 and VMSCL3. There are two OpenVMS VAX satellite nodes in the same cluster. They are named VMSS1 and VMSS2. The cluster may be accessed through DECnet using the cluster alias of VMSCL. Each of the satellite nodes of the cluster has 0 votes. Each of the non-satellite nodes has 1 vote. This is a total of three expected votes and requires that two of the three voting members be running for the cluster to be available. Each of the voting members of the cluster has DECnet installed and running for file transfers and network communication.

## Network Monitoring

OSM on one node of a network can monitor the accessibility of other nodes on the network. This verifies that everything from the network connection to the OSM monitor process is working on both nodes. When setting up network monitoring, the most efficient and reliable method is to have each node monitor the network connection on two other nodes. When one node of the network goes down, two nodes will report a problem communicating with that node. If a network connection fails, two nodes will report a problem communicating with each other. This makes it very easy to pinpoint the problem and fix it. If there are multiple problems with the network, or multiple nodes in the network go down, there will be reports about two different problems. This still makes it easy to identify and fix the problems quickly.

In an environment which includes both TCP/IP and DECnet, it is necessary to verify that at least one node in the network communicates with both TCP/IP and DECnet. In this example, all three of the voting members of the cluster have both TCP/IP and DECnet. These are the nodes that will monitor one system that has only DECnet and another system that has only TCP/IP.

When setting up network monitoring, do not monitor any node of the network that is expected to be shut down for extended periods of time. If this were done, then when that node is shut down, OSM will start reporting that as an event. Excessive reporting of expected events can cause the system manager to ignore the messages and miss important events.

In most network environments, there are some nodes that are less critical to normal production or rely on another node to be able to function. Since satellite nodes in a VMScluster require that there be other nodes in the cluster functioning, these nodes will be monitored by one of the voting members of the cluster. This means that if the satellite node goes down, only one node will report the event.

The following chart diagrams each node to be monitored. The important parameters for that node are included. The NETWORK_ID parameter is not listed for each node. It should be set to the same non-zero value on all nodes. It is also necessary to set the network password on each node. For a complete description of setting up network monitoring in OSM, see *Chapter 8: Network Event Monitoring*. The arrows indicate which nodes are monitoring each other. Lines with arrows on both ends are nodes that are monitoring each other, while lines with arrows on one end are nodes that are being monitored but are not monitor any other node.

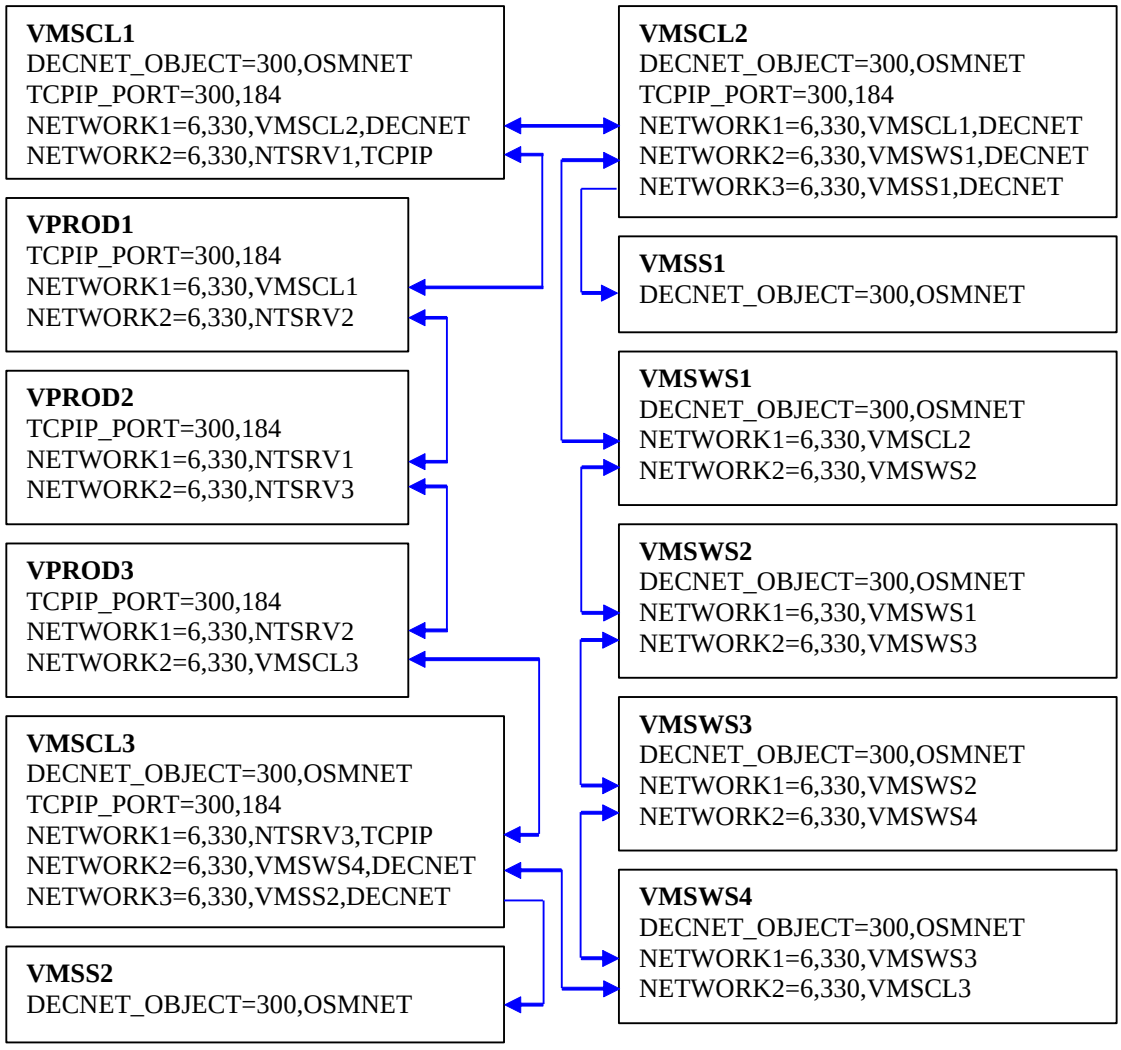



**Chart 1: Network Monitoring**

## OSM Transport Setup

Any one of the non-satellite nodes of the cluster, and each of the stand-alone systems, will do the reporting of all events reported by OSM to the system management staff. This reporting is done via mail. These systems will run the OSM Transport process.

The SMTP service in TCP/IP is configured to send email to the system manager. If those emails are on a separate domain than the systems, it may be necessary to enable mail relay on one of the corporate mail servers.

This chart displays each node and how each is set up. The reference to a complete node setup file indicates that the node setup file has information about all nodes being monitored by OSM. A short node setup file contains information about only the local node. The reporting enabled or reporting disabled flag indicates whether the reporting enabled check box is enabled in the complete node setup files for that node.

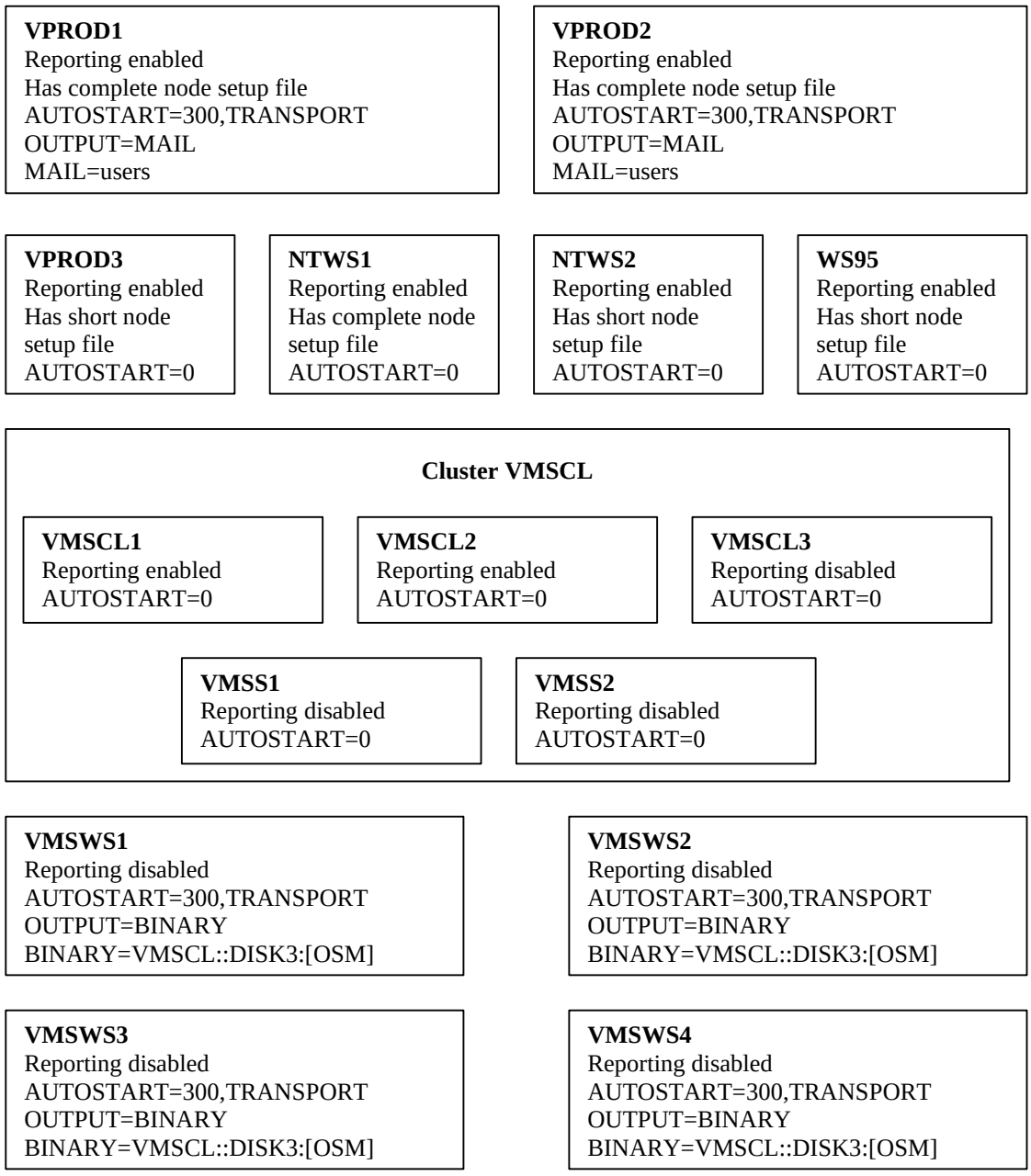



**Chart 2: OSM Transport Setup**

## Summary

The following table lists the parameters that are critical to this setup of OSM.

| Node Name | Notes |
|---|---|
| VPROD1 | This is one of the primary production servers.<br>Required parameter settings for transport setup:<br>OUTPUT=MAIL<br>MAIL=user-list<br>AUTOSTART=300,TRANSPORT<br>Required parameter settings for network monitoring:<br>NETWORK_ID=12345<br>TCPIP_PORT=300,184<br>NETWORK1=6,330,VMSCL1<br>NETWORK2=6,330,VPROD2 |
| VPROD2 | This is one of the primary production servers.<br>Required parameter settings for transport setup:<br>OUTPUT=MAIL<br>MAIL=user-list<br>AUTOSTART=300,TRANSPORT<br>Required parameter settings for network monitoring:<br>NETWORK_ID=12345<br>TCPIP_PORT=300,184<br>NETWORK1=6,330,VPROD1<br>NETWORK2=6,330,VPROD3 |
| VPROD3 | This is one of the primary production servers.<br>Required parameter settings for transport setup:<br>AUTOSTART=300,TRANSPORT<br>Required parameter settings for network monitoring:<br>TCPIP_PORT=300,184<br>NETWORK_ID=12345<br>NETWORK1=6,330,VPROD2<br>NETWORK2=6,330,VMSCL3 |
| VTEST1 | This system is used for trouble-shooting and experimental testing.<br>Required parameter settings for transport setup:<br>AUTOSTART=300,TRANSPORT |
| VTEST2 | This system is used for trouble-shooting and experimental testing.<br>Required parameter settings for transport setup:<br>AUTOSTART=300,TRANSPORT |
| VTEST3 | This system is used for trouble-shooting and experimental testing.<br>Required parameter settings for transport setup:<br>AUTOSTART=300,TRANSPORT |

| Node Name | Notes |
|---|---|
| VTEST4 | This system is used for trouble-shooting and experimental testing.<br>Required parameter settings for transport setup:<br>AUTOSTART=300,TRANSPORT |
| VMSCL1 | This is a voting member of the OpenVMS cluster. One of the voting members of the cluster runs the transport process to enable reporting.<br>Required parameter settings for transport setup:<br>AUTOSTART=300,TRANSPORT<br>Required parameter settings for network monitoring:<br>NETWORK_ID=12345<br>DECNET_OBJECT=300,OSMNET<br>TCPIP_PORT=300,184<br>NETWORK1=6,330,NTSRV1,TCPIP<br>NETWORK2=6,330,VMSCL2,DECNET |
| VMSCL2 | This is a voting member of the OpenVMS cluster. One of the voting members of the cluster runs the transport process to enable reporting.<br>Required parameter settings for transport setup:<br>AUTOSTART=300,TRANSPORT<br>Required parameter settings for network monitoring:<br>NETWORK_ID=12345<br>DECNET_OBJECT=300,OSMNET<br>TCPIP_PORT=300,184<br>NETWORK1=6,330,VMSCL1,DECNET<br>NETWORK2=6,330,VMSWS1,DECNET<br>NETWORK3=6,330,VMSS1,DECNET |
| VMSCL3 | This is a voting member of the OpenVMS cluster. One of the voting members of the cluster runs the transport process to enable reporting.<br>Required parameter settings for transport setup:<br>AUTOSTART=300,TRANSPORT<br>Required parameter settings for network monitoring:<br>NETWORK_ID=12345<br>DECNET_OBJECT=300,OSMNET<br>TCPIP_PORT=300,184<br>NETWORK1=6,330,VPROD1,TCPIP<br>NETWORK2=6,330,VMSCL2,DECNET<br>NETWORK3=6,330,VMSS2,DECNET |
| VMSS1 | This is a non-voting satellite member of the OpenVMS cluster.<br>Required parameter settings for transport setup:<br>AUTOSTART=0<br>Required parameter settings for network monitoring:<br>NETWORK_ID=12345<br>DECNET_OBJECT=300,OSMNET |

| Node Name | Notes |
|---|---|
| VMSS2 | This is a non-voting satellite member of the OpenVMS cluster.<br>Required parameter settings for transport setup:<br>AUTOSTART=0<br>Required parameter settings for network monitoring:<br>NETWORK_ID=12345<br>DECNET_OBJECT=300,OSMNET |
| VMSWS1 | This is a standalone OpenVMS workstation. This system sends OSM event data to the cluster for reporting.<br>Required parameter settings for transport setup:<br>OUTPUT=BINARY<br>BINARY=VMSCL::DATA8_DISK:[OSM]<br>AUTOSTART=300,TRANSPORT<br>Required parameter settings for network monitoring:<br>NETWORK_ID=12345<br>DECNET_OBJECT=300,OSMNET<br>NETWORK1=6,330,VMSCL2<br>NETWORK2=6,330,VMSWS2 |
| VMSWS2 | This is a standalone OpenVMS workstation. This system sends OSM event data to the cluster for reporting.<br>Required parameter settings for transport setup:<br>OUTPUT=BINARY<br>BINARY=VMSCL::DATA8_DISK:[OSM]<br>AUTOSTART=300,TRANSPORT<br>Required parameter settings for network monitoring:<br>NETWORK_ID=12345<br>DECNET_OBJECT=300,OSMNET<br>NETWORK1=6,330,VMSWS1<br>NETWORK2=6,330,VMSWS3 |
| VMSWS3 | This is a standalone OpenVMS workstation. This system sends OSM event data to the cluster for reporting.<br>Required parameter settings for transport setup:<br>OUTPUT=BINARY<br>BINARY=VMSCL::DATA8_DISK:[OSM]<br>AUTOSTART=300,TRANSPORT<br>Required parameter settings for network monitoring:<br>NETWORK_ID=12345<br>DECNET_OBJECT=300,OSMNET<br>NETWORK1=6,330,VMSWS2<br>NETWORK2=6,330,VMSWS4 |

| Node Name | Notes |
|---|---|
| VMSWS4 | This is a standalone OpenVMS workstation. This system sends OSM event data to the cluster for reporting.<br><br>Required parameter settings for transport setup:<br>OUTPUT=BINARY<br>BINARY=VMSCL::DATA8_DISK:[OSM]<br>AUTOSTART=300,TRANSPORT<br><br>Required parameter settings for network monitoring:<br>NETWORK_ID=12345<br>DECNET_OBJECT=300,OSMNET<br>NETWORK1=6,330,VMSWS3<br>NETWORK2=6,330,VMSCL3 |

# OSM Commands

The following are the OSM commands that control starting, stopping and waking the monitor and transport processes:

## DISABLE AUTOSTART

This command instructs the monitor process to no longer automatically restart one of the OSM processes. In a VMScluster, this command affects all monitor processes in the cluster. Once issued, autostart will be disabled as long as at least one node in the cluster is running the OSM monitor process or until autostart is re-enabled. This command will override the AUTOSTART parameter in the parameter file.

**Format**

**DISABLE AUTOSTART**

**Qualifiers**

**/TRANSPORT**

Specifies that autostart is to be disabled for restarting the transport process. This is the default if no qualifiers are specified.

## ENABLE AUTOSTART

This command will re-enable the autostart functionality. Once this command is executed all monitor processes that have the OSM parameter AUTOSTART enabled will automatically restart the processes specified by the AUTOSTART parameter if it is no longer running. This command does not enable autostart on nodes that have it disabled by setting the AUTOSTART parameter to 0.

**Format**

**ENABLE AUTOSTART**

**Qualifiers**

**/TRANSPORT**

Specifies that autostart is to be enabled for restarting the transport process. This is the default if no qualifiers are specified.

## SHOW STATUS

This command allows a user to see which version of OSM is executing on each node of the cluster, the PID of the OSM processes, the username that started each process, and a message if autostart has been disabled.

**Format**

**SHOW STATUS**

**Qualifiers**

**/FULL**

When combined with /SYSUAF, /FULL displays the name of each user record to be synchronized. If multiple changes have been specified, then the same user record may be displayed multiple times.

**/SYSUAF**

Includes a summary of SYSUAF modifications that OSM needs to notify another node or those that OSM needs to distribute to another node.

## START MONITOR

This command starts the OSM monitor as a detached process on the node on which the command is executed. The OSM monitor process collects the data to be sent by OSM transport process.

**Format**

**START MONITOR**

**Qualifiers**

**/[NO]LOG**

Specifying /LOG causes the command to display a one-line message showing the success of the command along with the process ID of the newly created process. When executed on the local node, this command displays the SHOW STATUS output after the monitor process has been started. /NOLOG suppresses all informational messages. The default is /LOG.

## START TRANSPORT

This command starts the OSM transport process on the local node. The transport process sends data collected by the monitor to location(s) specified in the parameter files. This command is normally not needed due to the autostart feature for the transport process.

**Note:** Only one transport process may be running in a VMScluster at a time. If a second transport process is started on the same node, the user will see the following message:

```
%SYSTEM-F-DUPLNAM, duplicate name
```

If a second transport process is started on a different node in the cluster, the OSM_TRANSPORT.LOG file will have the following error message:

```
%SYSTEM-W-NOTQUEUED, request not queued
```

**Format**

**START TRANSPORT**

**Qualifiers**

**/[NO]LOG**

Specifying /LOG causes the command to display a message showing the success of the command along with the process ID of the newly created process. When executed on the local PC, this command displays the SHOW STATUS output after the transport process has been started. /NOLOG suppresses all informational messages. The default is /LOG.

**/[NO]ENABLE_AUTOSTART**

Specifies that autostart should be enabled. This allows the user to enable the autostart feature for the transport process if it has been previously disabled.

## STOP MONITOR

This command shuts down the specified OSM process. Do not use the OpenVMS commands STOP PROCESS or STOP/ID to abort the OSM monitor process.

**Format**

**STOP MONITOR**

**Qualifiers**

**/CLUSTER**

Will cause all OSM monitor processes executing in the cluster to exit. The /NODE and /CLUSTER qualifiers are mutually exclusive.

**/[NO]LOG**

Specifying /LOG causes the command to display a one-line message showing the success of the command along with the process IDs of the affected processes. /NOLOG suppresses all informational messages. The default is /LOG.

**/NODE=node-name**

Allows the user to specify which node of the cluster on which the monitor process is to exit. The /NODE and /CLUSTER qualifiers are mutually exclusive. The default is to shut down the monitor process on the current node.

## STOP TRANSPORT

This command shuts down the OSM transport process. In a VMScluster environment, this command can be executed on any node in the cluster and will stop the transport regardless on which node it is running.

**Format**

**STOP TRANSPORT**

**Qualifiers**

**/[NO]LOG**

Specifying /LOG causes the command to display a one-line message showing the success of the command along with the process ID of the affected process. /NOLOG suppresses all informational messages. The default is /LOG.

**/[NO]DISABLE_AUTOSTART**

Specifies that autostart should be disabled. This allows the user to disable the autostart feature for the transport process at the same time the transport process is stopped to prevent the OSM monitor process from restarting it.

## WAKE TRANSPORT

This command is issued to wake up the OSM transport process to determine if there are files to be sent. If there are files to be sent, the OSM transport process then sends the files to the specified location. In a VMScluster environment, this command can be executed on any node in the cluster to cause the transport process to wake up.

**Format**

**WAKE TRANSPORT**

**Qualifiers**

**/[NO]LOG**

Specifying /LOG causes the command to display a one-line message showing the success of the command along with the process ID of the affected process. /NOLOG suppresses all informational messages. The default is /LOG.

# Environment Parameters

The following environment parameters control the basic operating environment of the OSM processes.

## AUTOSTART=300,TRANSPORT

This monitor process parameter enables or disables autostart for the node on which it is defined.

**Frequency**

The first value in the parameter list is the frequency, in seconds, which the monitor process will check for the existence of the processes specified as the second value in the parameter list.

**Process List**

The second value in the parameter list is a comma-separated list of processes that the monitor may start. Valid keywords are ALL, SERVER and TRANSPORT. Specifying ALL includes SERVER and TRANSPORT.

The default for a member of a VMScluster is AUTOSTART=0 which disables this functionality. This parameter should be set to AUTOSTART=300,TRANSPORT only on those nodes that should be running the OSM transport process. The default for a standalone OpenVMS system that is not part of a cluster, is AUTOSTART=300, TRANSPORT. If the parameter is entered, or defaulted, as AUTOSTART=300, TRANSPORT, once every 300 seconds (5 minutes) the monitor process will check to see if the transport is running somewhere in the cluster. If it is not, it will then be started on that node.

If this option is used, the OSM STOP command will stop one of the currently executing processes and the monitor process may restart it. This may be overcome with the DISABLE AUTOSTART command or the /DISABLE_AUTOSTART qualifier to the STOP command. Both are described in this chapter. The OSM installation procedure may put this parameter in the node-specific parameter file.

## BURST_DELAY=15

This monitor process parameter controls how long, in seconds, that the OSM monitor process will wait for additional events before sending a critical event. It is common for some events, such as device errors or login failures, to occur in bursts. This parameter allows OSM to wait a few seconds to gather all events in one burst and send them at one time.

## CPU_WAKE=300

This monitor process parameter controls how often the OSM monitor process evaluates rates for CPU modes and Split I/Os. Each time the OSM monitor process wakes up, it checks to see if this interval has expired. If it has, it calculates these rates. The purpose for having this parameter along with the MONITOR_WAKE parameter and the specified default values is to prevent short bursts of activity from causing false reports.

## CRITICAL=6

This sets the severity level of messages requiring immediate attention. All messages with this severity and higher will immediately be sent to the transport process for processing. See the *Errorlog Entry Thresholds* section in *Chapter 6: Monitoring for Exception Events* for information about how errorlog entries may also be treated as critical.

## EDITOR=TPU

This parameter specifies the editor used by OSM for editing the parameter files. This may be set to any editor which is supported by OpenVMS mail, or it can be specified as @file-spec to invoke a DCL command procedure for editing a file. This includes the EDT, TPU and TECO editors that are provided with VMS and the LSE editor which is a layered product available from DEC.

## HEARTBEAT=0

It is designed to periodically send messages to the locations that are receiving information from this system. The purpose for the heartbeat is to make sure that this node is alive and well. If the receiving locations do not hear from this node in the number of days specified by this parameter, there is some sort of problem. The problem may be a communication problem, OSM not running, or the OSM license has expired. A value of 0 disables this functionality.

## LNM_TABLE=LNM$FILE_DEV

This parameter determines which logical name table is used when defining and translating logical names. The default of LNM$FILE_DEV maintains compatibility with previous versions of OSM using any logical name in the default search list for the process. Placing the logical names in an OSM specific table keeps the system logical name table cleaner.

This parameter must be set the same in the parameter file and startup procedures.

## LOG=option

This parameter controls what information is logged by either the OSM monitor process or the OSM transport process in their log file. Each message included in a log file as specified by this parameter includes a time stamp and a message specifying what is being logged. Separating multiple keywords with a comma may enable more than one of these options. For example, LOG=FIX, PAGER will enable logging automated fix commands executed and pager strings which are generated by the OSM monitor process. By default, none of these options are enabled. The following table lists the available keywords.

| Keyword | Action |
|---|---|
| ALL | Specifying this keyword enables all other options listed below. It is possible to specify this keyword followed by the negated version of one of the following options. Example: LOG=ALL, NOFIX. |

| Keyword | Action |
|---|---|
| ANALYZE/ERROR | This causes the OSM transport process to log a message each time either the ANALYZE/ERROR, ANALYZE/ERROR/ELV, or DIAGNOSE command is executed in a subprocess to format OpenVMS errorlog data. |
| AUTOSTART | This causes the OSM monitor process to log a message each time it detects that the transport process is not running and automatically restarts it. |
| ERRORLOG | This causes the OSM monitor process to log a message each time that it executes the ANALYZE/ERROR, ANALYZE/ERROR/ELV or DIAGNOSE command in a subprocess to collect 24-hour OpenVMS errorlog data. |
| FIX | This causes the OSM monitor process to log the command string of all automated fixes executed by OSM in the OSM.LOG file. |
| PAGER | This causes OSM to log all PAGER notification strings that are included in the binary file for the transport process to be included in the OSM.LOG file. The pager notification string generated by the transport process, based on the severity of data detected, is always logged to the OSM_TRANSPORT.LOG file regardless of the setting of this parameter. |
| PAGER_SEND | This causes the OSM transport process to log a message in the OSM_TRANSPORT.LOG file each time that it invokes a page. |
| SYSUAF | This causes the OSM monitor process to log a message each time that it takes an action to synchronize user authorization files. This includes when it sends a message to another node to notify it of a change, when it distributes that change to other nodes, and when it is actually changing the user authorization files. |
| TIME_SET | This causes the OSM monitor process to log a message each time that it changes the time on the local node. This functionality is controlled by the NETWORK_TIME_SET# parameter. |
| UDM | This causes the OSM monitor process to log a message each time that it executes a command to execute a user-defined monitoring event in the context of a subprocess. |

## MONITOR_WAKE=30

This monitor process parameter specifies the wake-up interval for the OSM monitor process in seconds. This is how often the monitor wakes up to collect and analyzes data. This allows the system manager to control the amount of overhead the OSM monitor process generates. If this value is set very low, the process will put more of a load on the system but you will have faster reaction time. If it is too small, you could also log inaccurate rates when you have a small burst of high activity. If this interval is set too high, there will be a longer delay in reporting problems that OSM has discovered. The OSM monitor process will wake up immediately for security and error log events.

## NIGHT=00:30-05:00

This parameter defines a period of time for handling nightly processing. This parameter specifies a range of time, in the form of beginning time - ending time, specified as hh:mm-hh:mm. Nightly processing includes:

- The monitor process sends all non-critical events.
- The transport process restarts displaying repeated information based on day-of-week or day-of-month. For more information, see the REPEAT_DOM and REPEAT_DOW parameters.
- The transport process does automatic purge of the paging database. For more information, see the PAGER_DELETE parameter.

## OPCOM=N

This transport process parameter determines if an OPCOM message is sent to all operators' terminals whenever the transport process wakes up and determines that a binary event file needs to be processed. The valid options for this parameter are Y or N. If this parameter is enabled, an OPCOM message is sent each time the OSM transport process finds some files to process. The message is:

```
%OSM-I-SNDFIL, OSM is sending file(s)
```

If disabled, by a value of N, this OPCOM message is not sent. The messages are sent to the operator(s) specified by the OPERATOR parameter. The OPERATOR parameter is documented in Chapter 9, *Exception Event Reporting*.

## OPCOM_ERROR=1

This transport process parameter allows the user to decrease the number of messages that are reported when the transport process is unable to correctly process a file. The default value for this parameter, 1, causes all failed attempts to be reported to OPCOM. By increasing this value to a larger number, such as 3, messages will only be generated if OSM fails to correctly process at least 1 file on each of 3 consecutive attempts. The messages may be removed completely by specifying a value of 0. The messages are sent to the operator(s) specified by the OPERATOR parameter.

## PRIME=8:00-17:00,Mon,Tue,Wed,Thu,Fri

This parameter defines the data collection window, during which time the OSM monitor process will collect performance and capacity planning information.

**Time Range**

The first value in the list is a start-stop time range based on a 24-hour format. This is the time when OSM will collect performance and capacity planning information.

**Days of Week**

The rest of the values define which days to collect the data. The day may be entered as the days of week, with abbreviations as short as three characters being accepted, or as a number of 1 through 7 with 1=Sunday, 2=Monday, ... 7=Saturday. In the above example, OSM collects performance and capacity information from 8:00 a.m. until 5:00 p.m. Monday through Friday. See the CAPACITY parameter for additional information.

## PURGE_WORKING_SET=N

This parameter affects both the monitor and transport processes. If set to Y, both the monitor and transport processes will purge all memory from its' working sets prior to hibernating. When set to N, no working set purge is performed. This is most beneficial on systems that are very tight on memory.

## TRANSPORT_WAKE=300

This transport process parameter determines the frequency, in seconds, that the OSM_TRANSPORT process wakes up to send any files which need to be sent to the output destination(s), defined by the OUTPUT parameter. The default value of 300 seconds is equal to 5 minutes. The monitor process will wake up the transport process when it determines there is data to be processed. However, this is not done when the transport process on one system sends data to be reported to the transport process on another system via the OUTPUT=BINARY parameter. The transport will automatically be woken up each time the monitor process determines data should be sent immediately.

## UIC= [1,4]

This parameter controls the UIC of the OSM monitor and transport processes. Any legal UIC can be specified. Keep in mind that the UIC is comprised of OCTAL numbers. This allows the monitor and transport processes to be run using a UIC other than that of the process that starts OSM. This could be beneficial in setting up UIC based protection or UIC based identifiers to secure the data collected by OSM and thereby prevent access by unauthorized users. Specifying a UIC value of [0,0] causes the UIC of the process starting OSM to be used. The default is to use a UIC of [1,4].

This does not affect the username under which OSM is executing. OSM will always run with the username of the user that starts it. To have OSM start using a specific username, the OSM_STARTUP.COM file may be submitted to a batch queue using the /USERNAME=xxx qualifier. This is important because a transport process may try to send output to a binary file on another node of the network using a proxy account. If the username for the proxy account on the remote node does not match the username that started OSM, the network transfer will fail.

If the UIC parameter is changed, the monitor processes and the transport process on all nodes in the cluster should be stopped and restarted. Also, all nodes within the cluster should use the same UIC parameter. If the UIC is changed and one or more OSM processes are not restarted, or the OSM processes do not have the same UIC, you may get unpredictable behavior. If you modify this parameter, make sure the UIC you specify has read, write, execute and delete access to the OSM directories and files.

# Chapter 6
# Monitoring for Exception Events

## Operation

Monitoring is the major component of OSM. All other components may be invoked based on the information OSM collects while monitoring the system.

OSM collects information in two different ways. The first is by periodically waking up and collecting information on system performance, disk space utilization and memory usage. The OSM parameter MONITOR_WAKE determines how often the monitor process wakes up. Therefore, the minimum granularity which performance or disk space exception events can be detected is determined by this parameter.

The second is by establishing mailboxes that the operating system automatically writes security alarms, hardware error and software error information into. Since the monitor process is awakened immediately when any of these events occur, there is no need to poll OpenVMS for this information. OSM recognizes and processes these types of exception events immediately.

There are some events that can have very high rates for very short periods of time. If OSM were to collect information on these events during this period of time, it might look like an exception event even though it is normal expected behavior. To avoid erroneously reporting these events, OSM monitors them over a longer period of time to establish a realistic rate. The OSM parameter CPU_WAKE establishes the longer period of time to monitor these events.

The OSM process will wake up every MONITOR_WAKE interval and check for free space on disk(s), global pages, etc. It will then see if the CPU_WAKE interval has expired. If it has, OSM will collect rate information like page faults, CPU mode time, etc. If the CPU_WAKE interval has not timed out, it simply goes back to sleep.

Regardless of how the monitor process collects event data, it checks to see if the event data exceeds the user specified thresholds for that event; if so, this is called an exception event. All exception events are stored in an .ACTIVE binary event file. This file is kept in the main OSM directory and has a file name of OSM_DATAFILE_xxxx.ACTIVE, where the xxxx is the name of the node that is collecting data.

If an exception event occurred that has a severity level equal to or greater than the parameter CRITICAL, the OSM monitor process renames the .ACTIVE binary event file to a .SEND binary event file. The monitor process will also rename .ACTIVE binary event files to .SEND binary event files once a night if the .ACTIVE file exists.

Once a .SEND binary event file exists, the transport process discovers it the next time it wakes up and sends it to the designated destination(s) as defined by OSM parameters. For more information on .SEND binary event file processing, refer to *Chapter 9: Exception Event Reporting* in this manual.

Thresholds for events and what OSM should consider as an exception to normal system operation is defined by the OSM monitoring parameters discussed in the next section.

# Errorlog Entry Thresholds

To eliminate excessive errorlog entry reporting, OSM compares the number of errorlog entries it has received against a threshold. OSM will only report on errorlog entries as critical events when the number of entries exceeds the threshold for that device. The thresholds OSM uses to determine when to report on excessive error log entries are in a file that is in the main OSM directory named OSM.THRESHOLDS. The information for errorlog entries that have not reached the threshold is still included in the OSM reports to keep the user informed of all device errors reported on the system.

The thresholds set up in this file have been chosen after running OSM on several systems. The errorlog entry thresholds should not need to be modified. However, if they do need to be modified for your environment; for whatever reason, edit the OSM.THRESHOLDS file. It is a text file that has documentation within it to describe its format.

When the number of errorlog entries for a device reaches the specified threshold, the file will be sent as a critical file and the error count will be reset. If the number of errorlog entries received is not as large as the threshold for the associated device, the data will not be sent unless the severity of the event is deemed critical. All errorlog entries for which there is not a threshold are sent based solely on their severity. The errorlog entries that are compared against the thresholds are:

- Device Errors
- Device Time-outs
- Logged Messages
- Massbus Adapter (MBA) Errors
- Unibus Adapter (UBA) Errors
- Soft Error Correction Code (ECC) Memory Errors
- 8600 Corrected Read Data (CRD) Log
- Calypso Memory Scan Errors

Logged MSCP messages are not compared against the threshold file since they are usually generic messages from the controller and not the actual device. To eliminate a potential flood of error reporting when a device fails, a threshold is doubled each time a specified threshold is reached. The threshold is then reset each night back to the specified value.

For example, if the threshold for a particular disk is set to 4, once 4 errorlog entries for that disk are received within a single day, the OSM data collected will be sent as a critical event. For the next critical event to be triggered by that disk, 8 more errorlog entries must be received by OSM. This continues with 16, 32, 64 etc. until the following night when the threshold is reset back to the original value of 4 and the process repeats itself.

# Exception Event Monitoring Command

The following command may be used to assist in monitoring and setting up the monitoring parameters.

## SHOW TEMPERATURE

This command displays the current reading of the specified temperature sensor. This command is valid only for OpenVMS Integrity systems.

**Format**

**SHOW TEMPERATURE device-name**

**Parameter**

**device-name**

You must specify the name of the USB device representing the TEMPer sensor.

**Qualifiers**

**/CELSIUS**

Specifies that the temperature is to be displayed using the Celsius scale. This is the default and is the value as provided by the sensor..

**/FAHRENHEIT**

Specifies that the temperature is to be displayed using the Fahrenheit scale. By default, the temperature is displayed in Celsius.

# Exception Event Monitoring Parameters

These parameters control the collection and detection of OSM monitored exception events. They also define the normal (non-critical) and critical severity levels and the associated thresholds for exception events.

When defining the parameters, the first value is the severity associated with the specified threshold; the remaining values determine the thresholds. Valid severity levels are from 0 through 10. Specifying a severity level of 10 is considered by OSM to be the most severe and 1 is the least severe. Specifying a severity level of zero disables all functionality of that feature, including event reporting and paging. Specifying a severity level less than 0, or greater than 10, is invalid.

For many of these monitored events, there are two parameters: the parameter name for the monitored event and the parameter_C name for the monitored event. The parameter_C variant of each parameter sets the critical values while the other variant sets the non-critical values. Note that some of the monitored event parameters do not have a critical variant. For example:

**GBLSECTIONS=4,90,80**
**GBLSECTIONS_C=6,97,95**

GBLSECTIONS establishes the non-critical thresholds of 90% of global sections and 80% of global pages and assigns these thresholds a severity level of 4. GBLSECTIONS_C establishes the critical thresholds of 97% of global sections and 95% of global pages and assigns these thresholds a severity level of 6.

If a monitored event is triggered and the severity level associated with that event is equal to or exceeds the parameter CRITICAL, the event will cause the OSM_TRANSPORT process to send the information to all of its destination locations immediately. Unless modified, the critical level has been defined as 6.

All the parameters in this section are OSM monitor parameters. Therefore, if you change one of these parameter values you must stop and restart the OSM monitor process. For more information on stopping and restarting the monitor process, refer to *Chapter 5: Controlling OSM Processes* in this manual.

## AGE_ACCOUNTING=60 00:00:00

When OSM collects capacity planning data, it checks the age of the first entry in the system accounting file. If it is older than the delta time specified by this parameter, then it is included in the report.

This acts as a reminder to the system manager to maintain the size of this file. The default name of this file is SYS$MANAGER:ACCOUNTNG.DAT. A new file may be created with the SET ACCOUNTING /NEW_FILE command.

**Parameter Values**

- Age – specified as an OpenVMS delta time; the default is 60 days.

## AGE_ERRORLOG=60 00:00:00

When OSM collects capacity planning data, it checks the age of the first entry in the system errorlog file. If it is older than the delta time specified by this parameter, then it is included in the report.

This acts as a reminder to the system manager to maintain the size of this file. This file is named SYS$ERRORLOG:ERRLOG.SYS. A new file may be created by renaming the existing file.

**Parameter Values**

- Age – specified as an OpenVMS delta time; the default is 60 days.

## AGE_OPERATOR=60 00:00:00

When OSM collects capacity planning data, it checks the age of the first entry in the operator log file. If it is older than the delta time specified by this parameter, then it is included in the report.

This acts as a reminder to the system manager to maintain the size of this file. The default name of this file is SYS$MANAGER:OPERATOR.LOG. A new file may be created with the commands REPLY/ENABLE and REPLY/LOG.

**Parameter Values**

- Age – specified as an OpenVMS delta time; the default is 60 days.

## ALIGNMENT_FAULTS=2,1000

Alignment faults occur when the system references data which is not properly aligned on a natural memory address boundary. This situation is usually caused by application design and may hurt the performance of the application or system.

This parameter is only effective on OpenVMS Integrity systems and some OpenVMS Alpha servers.

**Parameter Values**

- Severity level – default is 2
- Alignment fault rate

The ALIGNMENT_FAULTS parameter sets the severity level and thresholds to report on excessive alignment fault rates. The above examples set the severity level to 2, and monitors for alignment faults at 1000 per second or greater.

Alignment faults are reported as part of the performance category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Alignment faults per second, severity 2
  kernel    exec   super    user
       0       0       0    1402 at 30-SEP-2011 14:18
```

## AUDIT=Y

OSM monitors security alarms on OpenVMS by watching the audit server process for messages. This parameter enables or disables this monitoring. When disabled, all other AUDIT_* parameters are ignored.

**Parameter Values**

- Enable / Disable flag – A value of Y enables monitoring the Audit Server process.

## Audit Event Parameters

The following table shows the Audit Server messages that may be reported by OSM and their default severity levels. Reporting on each of the Audit Server messages can be controlled by the use of parameters in the OSM control file. Each of the Audit Server message types has an associated parameter that will allow the user to determine the severity level. These parameters are only used if the AUDIT parameter is set to a value of "Y". To disable reporting on a specific Audit Server message type, assign the associated parameter to a value of zero.

| Audit Server Control File Parameter Names | Default Severity | Description |
|---|---|---|
| AUDIT_BREAKIN | 6 | Break-in attempts |
| AUDIT_BYPASS | 5 | Use of BYPASS privilege to access files and devices |
| AUDIT_INSTALL | 3 | Use of the install utility to add known image |
| AUDIT_LOGFAIL | 0 | Login failures |
| AUDIT_READALL | 4 | Use of READALL privilege to access files and devices |
| AUDIT_UAF | 5 | SYSUAF modifications other than a user changing their own password during login or with the set password command |

Security auditing events are reported as part of the security category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Security audit alarms, severity 3
      Date / Time      Type        Subtype       Node    Username      ID
18-MAR-2009 10:26:39 LOGFAIL    NETWORK          PAUL   DECNET       000000C5
```

The format of security events for the brief report matches that provided by the OpenVMS command ANALYZE /AUDIT /BRIEF. The format of the full report matches the ANALYZE /AUDIT /FULL command.

## BUGCHECK=9

This parameter sets the severity level of an OpenVMS fatal bugcheck as detected by OSM to a level of 9. These are the bugchecks that cause OpenVMS to go down and reboot. The ERRFMT process reports bugchecks that do not cause the system to go down. The severity of these bugchecks is controlled by the ERRFMT_BUGCHECK parameter. OSM detects system crash bugchecks through the OSM_SDA.COM command procedure which should be invoked at system startup.

**Parameter Values**

- Severity level – default is 9.

In order for OSM to determine whether a bugcheck occurred, the procedure SYS$MANAGER:OSM_SDA.COM must be invoked by either the SYSMAN startup database or the OpenVMS startup command procedure SYSTARTUP_VMS.COM. This procedure invokes SYS$MANAGER:OSM_SDA_COMMANDS.COM which may be modified to control the detail of information included in the report.

## CAPACITY=1

OSM collects capacity planning and tuning information during the time period specified by the PRIME parameter. This parameter sets the severity level of this information. The data collected includes average CPU mode times, XQP cache statistics, memory usage, memory pool usage, page/swap file usage, minimum disk free space and page fault rates. These averages are from the beginning of prime time or OSM monitor start time, whichever is later and continues to the end of prime time.

**Parameter Values**

- Severity level – default is 1.

Capacity planning information is reported as part of the capacity category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Capacity planning information, severity 1
Prime time is 12:30-13:30
First errorlog entry is at 20-JAN-2782 13:34

  int, kern, exec, super, user, comp, mp syn, null
    0%    1%    0%    0%    0%    0%    0%   97%

_PAUL$DKA0:           has  5.7% (233936 blocks) of 4110480 free
_PAUL$DKA100:         has  5.4% (158247 blocks) of 2940951 free
_PAUL$DKA200:         has 40.9% (3426156 blocks) of 8380080 free
_PAUL$DKA300:         has 19.8% (1656306 blocks) of 8380080 free
```

## COMPUTABLE=5,95,12,1
## COMPUTABLE_C=8,98,12,1

Processes which become compute-bound can impact the performance of the rest of the system. On a multi-processor system, it may take more than one compute-bound process before other uses notice the impact. This parameter monitors for compute-bound processes.

**Parameter Values**

- Severity level – default is 5 for non-critical and 8 for critical.
- Percentage of a single CPU used to classify a process as computable.
- Number of CPU_WAKE intervals process must exceed threshold to trigger event.
- Number of direct and buffered I/Os a process can do while being considered compute-bound.

The COMPUTABLE parameter sets the non-critical severity level and reporting thresholds for a compute-bound process. The COMPUTABLE_C parameter sets the critical severity level and thresholds. The above examples set the non-critical severity to 5, the threshold at 95% of a single CPU used for 12 CPU_WAKE intervals. They set the critical severity level to 8 and set the threshold at 98% of a single CPU used for 12 CPU_WAKE intervals. In both cases, if the process does any more than 1 I/O during the interval, it is not considered compute-bound.

The default value of 12 CPU_WAKE intervals with a default value of 300 for CPU_WAKE only identifies a process as compute-bound if it is that way for 1 hour or longer. Setting the threshold for I/Os to 1 allows the output of control/T or other I/O once each interval.

If there are more computable processes than there are processors on the system, then the threshold for the percentage of a single CPU used is adjusted down. For example, if there are 4 processors and 5 compute-bound processes, then a threshold of 90% of a single CPU is adjusted to 90% * 4 / 5 or 72%. This adjusts for how much time a process can receive when sharing with other processes on the system.

Computable processes information is reported as part of the performance category. See

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Compute-bound Processes, severity 9
     PID Name     Pri       CPU Time   Connect time
203B9654 WILLIAMS   4     0 00:09:54     0 00:11:31  at 26-FEB-2024 10:11
DSA2:[OSM]OSM_CONTROL_IA64_0840.EXE
```

This display shows the OSM_CONTROL_IA64_0840.EXE image being compute-bound. It was running the DEMO CPU command at the time. It was compute-bound for nearly 10 minutes while the process had been created 11 1/2 minutes earlier. Since it had a priority of 4, it could be impacting other users.

## CONFIGURE=Y, 300

OSM can monitor disks on the system for the amount of free space available for use and I/O rates. To do this, OSM must know which disks are available and mounted on the system. To get this information, OSM watches the system for new disks that become available after the OSM monitor process has been started. The CONFIGURE parameter determines how OSM watches for devices that are being dynamically added to the system.

**Parameter Values**

- Enable / Disable flag – if set to Y, OSM monitors the CONFIGURE process' mailbox. When the CONFIGURE process receives a message in the mailbox, OSM will rescan for available disks.
- Frequency in seconds for which OSM checks for additional disks.

The scan will detect devices that have been added to the system via the SYSGEN utility (OpenVMS VAX), the SYSMAN IO command (OpenVMS Alpha and Integrity Servers) the OpenVMS CONFIGURE process. Setting this to a value of zero disables the periodic scan.

The default value of CONFIGURE=Y,300 will cause OSM to scan the I/O database every 300 seconds and monitor messages from the operating system.

## CPU=1,20,35,10

OSM monitors the amount of time spent in the various CPU modes on the system. Each mode represents different portions of OpenVMS executing. An excessive amount of time spent by the system on the interrupt stack, in kernel mode or in multiprocessing synchronization may indicate a problem. This parameter sets up the severity level and thresholds to report on excessive time spent in the various CPU modes. The thresholds represent the percentage of total CPU time spent in each mode.

**Parameter Values**

- Severity level – default is severity 1.
- Interrupt stack time – percentage of time spent processing device interrupts and handling scheduling activities. The default is 20%.
- Kernel mode time – percentage of time spent in kernel mode. Kernel mode is used by the system when initiating I/O, and processing file system and lock requests. The default is 35%.
- Multiprocessing synchronization time – percentage of time spent requesting a spinlock for synchronization in a multi-processor system. The default is 10%.

CPU utilization is monitored each CPU_WAKE interval (default 300 seconds) to avoid short spikes from falsely triggering an exception event. The report for CPU utilization provides the same information displayed by the DCL command MONITOR MODES.

CPU utilization is reported as part of the performance category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** CPU usage by mode, severity 1
  int, kern, exec, super, user, comp, mp syn, null
   40%   21%    1%    0%   14%    0%    0%   24% at 23-APR-2009 08:25
Available cpu count = 00000002, Active cpu count = 00000001
```

The first column is the percentage of time spent on the interrupt stack. Typically, the CPU spends time on the interrupt stack when performing device I/O completion and distributed lock management. In a VMScluster, it may be substantially higher for systems serving disks. Too much time spent on the interrupt stack may be an issue to be investigated.

The second column is the percentage of time spent in kernel mode. Kernel mode time is typically time spent performing paging, swapping, lock management, other OpenVMS internal routines or the file system (XQP). Time spent here is usually considered overhead, which is something you would like to minimize. If there is too much time spent in kernel mode, you may have a performance bottleneck somewhere other than the CPU.

The next column is the percentage of time spent in executive mode; this is time either spent in RMS or in a database product. This is usually considered productive time. If you spend a lot of time in executive mode, you may want to investigate tuning your RMS files or databases.

The next column is the percentage of time spent in supervisor mode, which is the time spent executing DCL commands or command procedures. Excessive time spent in this mode is not efficient. If this value is high, convert the DCL command procedures to programs written in high level languages. Programs written in high level languages are typically more efficient than DCL command procedures.

The next column is the percentage of time spent in user mode time, the most desirable time. This is typically where applications spend most of their time. If you have a high percentage of time spent in user mode, you have a well-tuned system. If the CPU is 100% busy and most of the time is spent in user mode, you probably have a CPU bottleneck. Consider adding processors to the system, adding systems to the cluster, or reviewing the application to increase CPU capacity or decrease application requirements.

The next column is the percentage of time spent in compatibility mode time; this is time spent emulating a PDP11/XX machine. Only older VAX systems are capable of performing this action.

The next column is multiprocessor synchronization time. In a multiprocessor, the CPUs must synchronize access to memory resident databases like the scheduler database. This time is considered overhead, as applications are not able to run during that time.

If a multi-processor system is regularly spending more than one full CPU in multi-processor synchronization, and it is handling a large number of lock requests, it may be possible to improve performance by enabling the dedicated CPU lock manager. This dedicates 100% of a single CPU to processing lock requests while eliminating the synchronization required to access lock manager data structures in memory.

In general, what you want is more time in user and executive mode combined, than in interrupt and kernel. In non-clustered systems, anything above 30% in kernel and interrupt time may be too high.

A second line is displayed when some of the available CPUs are not running. This can happen if there is a hardware failure or if someone executes the STOP /CPU command.

## DEADLOCK=4,1,2
## DEADLOCK_C=6,2,20

Deadlocks are multiple processes, each waiting for a resource that is held by the other process. This situation is usually caused by application design. OSM monitors for deadlock searches and deadlock detections and reports an event each time the threshold is exceeded. Each occurrence is reported even if it is not as serious as the last one reported.

**Parameter Values**

- Severity level – default is 4 for non-critical and 6 for critical.
- Deadlock detection rate
- Deadlock search rate

The DEADLOCK parameter sets the non-critical severity level and thresholds to report on excessive deadlocks or excessive deadlock search rates. The DEADLOCK_C parameter sets the critical severity level and thresholds for deadlocks. The above examples set the severity level to 4, and monitors for either deadlock detections at 1 per second or greater, or deadlock searches initiated at 2 per second or higher. They set the critical severity level to 6, and monitors for 2 or more deadlock detections per second or 20 or more deadlock searches initiated per second.

Deadlocks and searching for deadlocks are reported as part of the performance category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Lock manager deadlock statistics, severity 6
Deadlock search rate reached 2 per second at 18-MAR-2009 10:36
Deadlock detection rate reached 2 per second at 18-MAR-2009 10:36
```

OpenVMS initiates a deadlock search whenever a process must wait more than the number of seconds specified by the DEADLOCK_WAIT system parameter. If OpenVMS detects a deadlock, it prevents the deadlock from causing application hangs by reporting to one of the processes involved that the requested resource is unavailable. This search for a deadlock is extra overhead for the system.

## DISK=2,95,200
## DISK_C=7,99,32000

Operating systems, and many applications, have problems when a disk they are using gets full. OSM monitors each mounted disk on the system for the amount of free space and excessive I/Os to the disk.

**Parameter Values**

- Severity level – default is 2 for non-critical and 7 for critical.
- Percentage of disk space used
- Number of I/Os per second

The DISK parameter sets the non-critical severity level and thresholds to report on disk space usage and I/O rates. The DISK_C parameter sets the critical severity and thresholds for these events and the severity of mount verification events. The above example sets the non-critical severity to 2; the threshold for used disk space at 95%, and the I/O threshold at 200 I/Os per second. It sets the critical severity level to 7, the threshold at 99% of disk space used, and 32000 I/Os per second. These values are applied on an individual disk basis.

When a disk gradually fills up, OSM reports each time the threshold is reached and when free space decreases by the larger of 10% of free space or 200 blocks. This prevents OSM from creating a flood of messages on a single disk that is slowly filling up.

The critical and non-critical parameters for disk monitoring can be overridden on a per disk basis through the use of logical names. The name is in the form of OSM_DISK_$1$DGA101 95,99,40,300,2,7. This specifies that the parameters for the disk with the physical name of $1$DGA101 should be 95% used for notification, 99% used for critical, 40 I/Os per second for notification, and 300 for critical. It sets the non-critical at severity 2 and critical at severity 7. It may specify a pager string or information for automated fixes. See *Chapter 10: OSM Paging Support* or *Chapter 11: Automated Fixes* for more information on these features.

OSM_LOGICALS.TEMPLATE contains an example of logical names for individual disks. Refer to *Chapter 5: Controlling OSM Processes* for a more detailed description.

Disk utilization is reported as part of the performance category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Disk usage statistics, severity 2
_PAUL$DKA100:          88842 of 2940951 free at 23-APR-2009 08:18
        with 1150 i/o's per second at 23-APR-2009 08:23
```

The amount of free space needed on a disk is highly dependent on the applications using that disk. A database server that preallocates storage for the primary database files may not be affected at all by a full disk. However, the disk containing the journal files for that database may require significantly more space to store the on-going transactions.

A high I/O rate to one disk can cause a performance problem for any application accessing that disk. This can frequently be remedied by moving some of the more heavily accessed files from one disk to another. A disk or file caching utility may allow the I/O rates to a disk to get very high. In this case, adjust the DISK or DISK_C parameters to not report the high I/O rates as an exception event by setting the rate of I/Os per second to 32000.

## ERRFMT=Y

This parameter is used by OSM to determine if it should monitor events that come from the OpenVMS error format process. Disabling the monitoring of messages from the ERRFMT mailbox causes OSM to ignore all of the ERRFMT_* parameters.

**Parameter Values**

- Enable / Disable flag – if set to Y, OSM monitors the ERRFMT mailbox for errors.

**Note:** OpenVMS V8.2 through OpenVMS V8.3, on both Alpha and Integrity, require the installation of the latest ERRFMT patch. Without this patch, the ERRFMT process sends corrupted data to OSM. The fix has been included in OpenVMS V8.3-1H1.

Device errors are reported as part of the hardware category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** ERRFMT information, severity 8
Error Log Summary Report

Summary of all entries logged
      1 User bugcheck

Date of earliest entry:  23-APR-2009 08:40
Date of latest entry:    23-APR-2009 08:40
```

The format of the data in this report varies greatly depending on the tool used to format the data. The above example was generated by OSM. The parameters to control the formatting are ERRORLOG_FORMAT and ERRORLOG_TRANSLATE.

## Error Type Parameters

The following table shows which error format types may be reported by OSM and the default severity associated with each message. If error format reporting is enabled, reporting on each of the individual error format types can be controlled by the use of these parameters and may be disabled by assigning the associated parameter the value of zero.

| Error Format Control File Parameter Names | Default Severity | Description |
|---|---|---|
| ERRFMT_BUGCHECK | 8 | Nonfatal Bugcheck Errors |
| ERRFMT_BUS | 6 | External Bus Errors |
| ERRFMT_CARD | 1 | Card Reader Errors |
| ERRFMT_DISK | 6 | Disk Errors |
| ERRFMT_HARD_MEMORY | 6 | Hard Memory Errors |
| ERRFMT_INFORMATION | 0 | Informational Errors |
| ERRFMT_LINE_PRINTER | 0 | Line Printer Errors |
| ERRFMT_MAILBOX | 1 | Mailbox Errors |
| ERRFMT_MEMORY | 4 | Memory Errors |
| ERRFMT_MISCELLANEOUS | 1 | Miscellaneous Device Errors |
| ERRFMT_POWER | 5 | Power Fail Recoveries |
| ERRFMT_REALTIME | 7 | Real-time Device Errors |
| ERRFMT_SYNCHRONOUS_ COMMUNICATION | 4 | Synchronous Communication |
| ERRFMT_SYSTEM | 6 | CPU, Console and Bus Errors |
| ERRFMT_TAPE | 2 | Tape Errors |
| ERRFMT_TERMINAL | 0 | Terminal Errors |
| ERRFMT_VOLUME | 1 | Volume Mounts and Dismounts |
| ERRFMT_WORKSTATION | 4 | Workstation Errors |

## GBLSECTIONS=4,90,80
## GBLSECTIONS_C=6,97,95

Global sections are used in OpenVMS to share data among multiple processes. The data may be an image which has been installed with the INSTALL utility, file information shared by RMS in the form of global buffers, or a global section created by an application to share with other users of that application. Each global section contains one or more global pages. If OpenVMS runs out of either of these resources, the consequences could be very minor, or it could mean the failure of a critical application.

**Parameter Values**

- Severity level – default is 4 for non-critical and 6 for critical.
- Percentage of global sections used
- Percentage of global pages used

The GBLSECTIONS parameter sets the non-critical severity level and thresholds for reporting on global page and global section availability. The GBLSECTIONS_C parameter sets the critical level and thresholds for these resources. The above example establishes the non-critical thresholds of 90% of global sections used and 80% of global pages used and assigns these thresholds a severity level of 4. The critical threshold of 97% of global sections used and 95% of global pages used are associated with a severity level of 6.

Information regarding the use of global sections is reported as part of the performance category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Global section usage statistics, severity 6
Global sections -- 15 of 830 free at  1-MAY-2009 11:32
Global pages -- 779913 of 807606 free at  1-MAY-2009 10:24 with 364464 contiguous
```

In many cases, the amount of global sections and pages that is available is relatively static. However, some applications will create and use global sections/pages. If the number of free global pages or sections were too small, those types of application may fail, or you would not be able to use the OpenVMS INSTALL utility to install shared images or data files.

When global sections or global pages are low, the solution is to increase either the system parameter GBLSECTIONS or GBLPAGES. The system must be rebooted before these changes will take effect.

## IO=3,20,10
## IO_C=8,32000,32000

A large number of split I/Os or window turns is usually an indication of disk fragmentation. While this usually does not cause an application failure, it does affect overall system performance. The fragmentation on the disk forces OpenVMS to do multiple I/Os to the disk to retrieve a single piece of information. Direct and buffered I/O rates, along with the number of file opens, are controlled by mostly by application design.

**Parameter Values**

- Severity level – default is 3 for non-critical and 8 for critical.
- Split I/O rate
- Window turn rate

The IO parameter sets the non-critical severity level and thresholds for detecting excessive split I/O and window turn rates. The IO_C parameter sets the critical severity level and thresholds, along with the severity assigned to disk mount verifications. The above example sets the non-critical severity to 3, and causes OSM to report an exception event when the split I/O rate reaches 20 per second or the window turn rate reaches 10 per second system-wide. It also sets the critical severity to 8, and monitors for 32000 split I/Os per second and 32000 window turns per second system-wide. This effectively eliminates reporting any critical levels of window turns or split I/Os. If you feel that excessive window turns or split I/Os are critical to your site, modify this parameter accordingly. These rates are checked every CPU_WAKE interval (default 300 seconds) to avoid short spikes from falsely triggering a disk fragmentation message.

I/O rates are reported as part of the performance category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** System I/O counts, severity 3
Split I/Os per second reached 5 at 29-SEP-2011 23:59
Buffered I/Os per second reached 670 at 30-SEP-2011 08:35
Direct I/Os per second reached 776 at 29-SEP-2011 16:43
Maximum number of files open reached 835 at 30-SEP-2011 10:31
File opens per second reached 137 at 30-SEP-2011 08:35
Window turns per second reached 13 at 30-SEP-2011 08:35
Erase I/Os per second reached 3 at 29-SEP-2011 16:43
```

This report displays direct and buffered I/O rates, split I/Os, window turns, erase I/O rates, and the rate of file open operations. Those operations with a rate of 0 per second are not included in brief formatted reports.

Split I/Os occur when files are fragmented. The operating system has to "split" a single I/O into multiple I/Os to get information that is spread across multiple fragments. A high split I/O rate indicates you have a severe fragmentation problem. Backing up and restoring the affected disk is one way of curing fragmentation. Using a disk defragmentation product, such as Disk File Optimizer from HP or one of the available third-party products, is another. Keep in mind that most defragmentation products can not defragment open files.

Buffered I/Os are usually terminal and network I/Os. This information will help you understand what is happening in your system; but without access to the source code for the product creating a high buffered I/O rate, it is difficult, if not impossible, to fix the problem.

Direct I/Os are normally disk and tape I/Os. High direct I/O rates to a disk may indicate that the XQP or XFC caches need tuning, you must load balance files or you might have a fragmented disk.

You really can not do much if the maximum number of open files, file open per second rates, or the erase I/O rates are excessive. If the file opens or erase I/Os are excessive, check that the XQP caches are at least 85% effective. If the erase I/O rate is excessive, make sure that you have turned off high-water marking and the erase on delete feature on your disks if those features are not needed.

## LOGICAL_NAMES=1,65,0,6

OpenVMS uses a hash table to speed the translation of logical names. This table breaks up shared logical names into groups. When translating a logical name, OpenVMS goes directly to the correct group containing the translation. It then searches sequentially through that group to find the logical name. The size of the hash table affects the size of each group in the hash table. If the hash table is too large, some system memory is lost. If it is too small, each group in the hash table can become very large; requiring an excessive amount of time being spent scanning through the list for a logical name.

Since OpenVMS and many applications have several logical names that are created in the job logical name table and are the same name for each user, some of the hash tables can grow large based on the number of jobs on the system. Two of these logical names are SYS$LOGIN and SYS$SCRATCH.

**Parameter Values**

- Severity level – default is 1.
- Percentage of shared logical name table unused
- Largest number of logical names in a single hash table – a value of 0 defaults to the system parameter MAXPROCESSCNT.
- Average length of hash table

The above example sets the severity level to 1, the percent of unused shared logical name hash table entries to 65%, and the average length of the hash chains to 6 logical names.

The use of logical name hash tables is reported as part of the performance category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Logical name table usage, severity 1
Effective LNMSHASHTBL is 512 with 68% used.  1094 logical names.
Maximum number of logicals in one hash chain is 73.
Average number of logicals in each hash chain used is 3.
```

This report displays logical name table hashing effectiveness. Logical names are stored in logical name data structures. To keep track of the logical name data structures, a hash table is created based on the SYSGEN parameter LNMSHASHTBL. Each entry in the hash table is a list head pointing to a list of logical name data structures. When translating logical names, OpenVMS uses a formula, called a hashing algorithm, to locate the proper list head, called a hash chain. It then has to search through each logical name data structure to see if that data structure contains the logical name that is being translated. The more logical name data structures on one hash chain, the slower the translation creating more system overhead (more time in kernel mode). For optimum performance, there would be few logical name data structures per hash chain.

In this example 68% of the lists were populated. The average hash chain has 3 logical name data structures, and the largest chain has 73 logical name data structures. If a hash chain gets too populated, increasing the system parameter LNMSHASHTBL may help improve performance.

## PAGE_FAULTS=2,500,40,2

Page faults are a normal routine in any virtual memory system, but too many page faults may be an indication of excessive overhead that might possibly be reduced. OSM monitors the overall number of page faults on the system. To prevent a short spike of page faulting from causing an exception event, OSM will only monitor page fault activity at a frequency as defined by the CPU_WAKE parameter.

**Parameter Values**

- Severity level – default is 2.
- Overall system-wide page faults per second
- System-wide hard fault rate per second
- System fault rate per second

In the above example, the severity level is 2. The second value specifies an overall system page fault threshold of 500. This includes all process/system faults both, hard and soft page faults. Hard page faults are those that require an I/O to complete. These are the faults that cause the whole system to wait not just one process. The system fault rate is the number of page faults incurred by the operating system itself.

Page faults are reported as part of the performance category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Page faults per second, severity 2
faults rdio wrio free  mod  dzro gvld wip sys  fpl    mpl
  1748 252   0     9    55  1378   54   0  0  13916 13675 at 30-SEP-2011 08:35
```

The brief display for this output includes the main fields displayed by the OpenVMS utility MONITOR. The full report adds the collided and release-pending page fault rates.

A non-zero system fault rate should be reviewed for possible correction. If the system fault rate is high, increasing the system parameter SYSMWCNT usually resolves the issue. Changing process page faulting is usually a matter of adjusting working sets and other memory usages to get the best balance of performance on the system.

The first column in the display (faults) is the overall page fault rate for the entire system. This number includes all process and system hard and soft page faults. A reasonable rate is very dependent on the system and hardware configuration, as well as the applications being executed.

The second column (rdio) is the number of times OpenVMS must go to disk to resolve a page fault, which are considered hard page faults. Obviously, you want to minimize the hard fault rate as much as possible since disk I/O is very time consuming.

The third column (wrio) is the number of write I/Os the swapper process is performing to write out the pages on the modified page list. After the pages are backed up, they are moved to the free page list. To alleviate this, you can either increase the working sets of the processes generating the faults, or increase the size of the modified page list. Which option you choose depends on the number of free pages that are available on your system. The SYSGEN parameter MPW_HILIMIT controls the maximum size of the modified page list.

The fourth through eighth columns are different types of soft faults. Soft faults do not require an I/O to complete the page fault. The fourth column (free) indicates faults resolved from the free page list. To alleviate a high free page fault rate, you can increase the working sets of the processes generating the faults if your system has enough free pages.

The fifth column (mod) indicates page faults resolved from the modified page list. To alleviate a high modified page fault rate, you can increase the working sets of the processes generating the faults if your system has enough free pages.

The sixth column (dzro) indicates demand zero-page faults (i.e. give me a page of 0s). You typically see demand zero faults during image activation. There really isn't anything you can do to alleviate these, but you may want a larger free page list or consider raising the system parameter ZERO_LIST_HI.

The seventh column (gvld) indicates page faults for which the data is already in physical memory in a shareable (global) image. Each time you see a global valid fault, your system is saving a page of memory and typically an I/O request. Global valid faults aren't considered bad, but excessive global valid faults may indicate that users working sets are too small. To alleviate a high global valid page fault rate, you can increase the working sets of the processes generating the faults if your system has enough free pages.

The eighth column (WIP) identifies page faults that are resolved by reclaiming data that is currently being written to the page file as part of column three. This is probably the worst of the faults. To alleviate this, you can either increase the working sets of the processes generating

the faults, or increase the modified page list. Which option you choose depends on the number of free pages that are available on your system.

The ninth column (sys) indicates system page faults, which slow down not just one individual, but the whole system. To alleviate a high system fault rate, increase the SYSGEN parameter SYSMWCNT.

The next two columns list the size of the Free Page List and the Modified Page List. Except for the last two columns, all this data is displayed in rates per second.

## PAGEFILE=3,65,70
## PAGEFILE_C=8,85,90

OpenVMS uses page and swap file space to store portions of an application that can not be held in physical memory. If the page files get too full, there is no other place to put those portions of a process that do not fit in physical memory. If the swap files are full, and OpenVMS needs to swap out another process, it swaps that process out to a page file. Once this happens, system performance can degrade significantly. OSM monitors both page and swap file usage to provide an early warning of the problems before they occur. When OSM detects that total page or swap file space has increased, it considers any existing limitations resolved.

**Parameter Values**

- Severity level – default is 3 for non-critical and 8 for critical.
- Percentage of page file space used
- Percentage of swap file space used

The PAGEFILE parameter sets the non-critical severity level and thresholds for page and swap file usage. The PAGEFILE_C parameter sets the critical severity and thresholds. The above example sets the non-critical severity level to 3 and monitors for more than 65% of the page file or 70% of the swap file being used. It sets the critical severity to 8 and monitors for more than 85% of the page file or 90% of the swap file used.

Page and swap file usage is reported as part of the performance category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Page and swap file statistics, severity 8
Page file 90% (634176 blocks) of 704640 used at 18-MAR-2009 12:07
```

The only short-term solution to resolving an issue with running out of page file or swap file space is to create an install another file using the SYSGEN utility. For a long-term solution, you may either increase the size of the existing page or swap files, or install an additional file.

## POOL=3,30,65
## POOL_C=8,85,90

Paged and nonpaged memory pools are critical system resources, which if depleted, may cause the system to hang or cause application or network communication failure. Paged pool contains some of the file system caches, shared logical names and other resources. Nonpaged pool contains device driver data structures, lock blocks, I/O requests, and other resources. When paged pool gets full, any additional requests for it will fail. When nonpaged pool is full, OpenVMS attempts to expand it. System parameters control the size of the sections of pool and how far nonpaged pool is allowed to grow. OSM monitors the usage of pool and its growth and reports when there is too little free paged pool or too little room left for growth in nonpaged pool. OSM monitors allocation failures from the paged and nonpaged pools.

**Parameter Values**

- Severity level – default is 3 for non-critical and 8 for critical.
- Percentage of non-paged pool expansion space used
- Percentage of paged pool used

The POOL parameter sets the non-critical severity level and reporting thresholds for nonpaged and paged pool usage. The POOL_C parameter sets the critical severity level and thresholds. The above examples set the non-critical severity to 3, the threshold at 30% of nonpaged pool expansion area used, and 65% of paged pool used. They set the critical severity level to 8 and set the threshold at 85% of nonpaged pool expansion area used, and 90% of paged pool used.

The POOL_C parameter controls reporting of paged and nonpaged pool allocation failures. These parameters do not have a value setting for allocation failure thresholds. If OSM detects an allocation failure for paged or nonpaged pool, it is reported as a critical event. Setting the severity level of POOL_C to 0 disables the reporting of the allocation failures.

Paged and non-paged pool usage is reported as part of the performance category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Memory pool statistics, severity 3
Memory pool usage first detected after uptime of 0 00:05:06
NPAGEDYN is 1998848, using 3989504 at 22-APR-2009 08:42 with 0 allocation failures
PAGEDYN is 20% (244496 bytes) free at 22-APR-2009 08:48 with 0 allocation failures
```

This display shows the initial size as the first number and the current size as the second number for nonpaged pool that has expanded. If the second number is larger than the first number, you have had expansion, which in the majority of cases is not good.

Another portion of the report points out the number of allocation failures. An allocation failure is when a request is made to an area of nonpaged pool that has been exhausted. If a request is made to the nonpaged pool and that area has been exhausted, severe problems can occur, and much more attention needs to be diverted to this event.

If you get this exception event information, set the parameter PAGEDYN or NPAGEDYN to what it has grown to in the MODPARAMS.DAT file, and execute AUTOGEN. A reboot is required to implement this change.

## POWER=5

Some systems have the ability to maintain the status of the CPU and memory during short power failures. These systems may then recover from such failures and resume normal operation. However, it may be beneficial to know when these occur and how long power was down. OSM monitors the system for the recovery of power failures and reports on them.

**Parameter Values**

- Severity level – default is 5.

The POWER parameter assigns the severity level to OSM detected power fail recoveries. If the system successfully handles a power fail and recovery, the OSM monitor is notified and makes an entry using this severity level. OSM reports the number of power fail recoveries it detects and the total amount of downtime due to those power failures.

## PROCESSES=5,80,85
## PROCESSES_C=8,90,95

OpenVMS allocates some memory resources for process slots based on the system parameter MAXPROCESSCNT. This determines the number of processes that may exist on the system at the same time. When these are all used, any attempt to create another process encounters the following error.

```
%SYSTEM-F-NOSLOT, no PCB available
```

The number of interactive processes allowed on the system is limited to prevent the creation of other production related processes. When this limit is reached, only users with the OPER privilege are allowed to log in interactively. Other users may see the following message while logging in. Once logged in, the user's process is not affected.

```
Maximum number of users exceeded; try again later
```

**Parameter Values**

- Severity level – default is 5 for non-critical and 8 for critical.
- Percentage of process slots used
- Percentage of interactive job limit used

The PROCESSES parameter sets the non-critical severity level and reporting thresholds for process availability. The PROCESSES_C parameter sets the critical severity level and thresholds. The above examples set the non-critical severity to 5, the threshold at 80% of process slots used, and 85% of the interactive job limit used. They set the critical severity level to 8 and set the threshold at 90% of process slots used, and 95% of the interactive job limit used.

Process availability information is reported as part of the software category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** Process availability, severity 8
71 of 148 process slots available at 18-JAN-2024 12:40
0 of 5 interactive logins available
146 peak processes on system
```

This display shows the initial size as the first number and the current size as the second number for nonpaged pool that has expanded. If the second number is larger than the first number, you have had expansion, which, in most cases, is not good.

If you get this exception event information, determine whether the event is caused by process slots or interactive jobs. If process slots, then increase the parameter MAXPROCESSCNT in the MODPARAMS.DAT file, and execute AUTOGEN. This change may require additional memory on the system. A reboot is required to implement this change. If the reported event is due to the interactive job limit, that may be increased with the command SET LOGIN /INTERACTIVE=###.

## TEMPERATURE=(no default)
## TEMPERATURE_C=(no default)

OSM supports monitoring temperature using the PCSensor TEMPer sensor. With this USB sensor, it is possible to monitor and alert when server room or rack temperatures exceed a threshold. Sensors can be found at http://pcsensor.com/.

**Parameter Values**

- Severity level
- Temperature
- USB device name – for example UGA1
- Scale – default is Celsius, may specify Fahrenheit
- Description – default is device name

When the sensor is plugged into an OpenVMS Integrity system, a UGAn device is created. This device name must be specified in the parameter to be monitored. Multiple sensors may be monitored by entering the parameter multiple times with different device names.

By default, the temperature entered and the temperatures being displayed in the report use the Celsius scale. This may be changed to Fahrenheit. If both a critical and non-critical entry is made for the same device, both parameters must use the same scale.

A description may be included in the report by specifying it in the parameter. If a description is not supplied, OSM uses the device name as the description.

Use the command SHOW TEMPERATURE to view the current temperature on the sensor.

High temperature events are reported as part of the hardware category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

To determine the current temperature, the SHOW TEMPERATURE command is executed.

```
$ osm show temperature uga1 /fahrenheit
Temperature 80.948044F
```

Then the following parameters were placed in the node specific parameter file setting the warning threshold at 85 degrees and critical at 95 degrees.

```
temperature=5,85,uga1:,f,primary server room
temperature_c=9,95,uga1:,f
```

After restarting the monitor process, the following event was generated.

```
****** Temperature events, severity 5
Temperature 90.95F primary server room  at 11-OCT-2011 16:28
```

## XQP_CACHE=3,4,90

The OpenVMS file system is the eXtended Qio Processor (XQP). To help its performance, it maintains a set of cache buffers. These buffers hold data that is normally kept on disk. These buffers improve performance by eliminating the need for the XQP to access the disks as frequently. If the buffers are sized too small, they are not as affective. This can hurt system performance. OSM monitors the effectiveness of the cache and reports an exception event any time that the hit rate of the cache is low while the attempt rate is high.

**Parameter Values**

- Severity level – default is 3.
- Minimum attempt rate per-second to report
- Hit rate percentage

This parameter determines the severity level and reporting threshold for XQP cache effectiveness. The above example sets the severity level to 3. It also specifies that the required attempt rate must be greater than or equal to 4 and the cache hit percentage be less than 90% before OSM will report on cache effectiveness.

XQP cache statistics are reported as part of the performance category. See discussion on

Output Groups in *Chapter 9:*
Exception Event Reporting for additional information.

**Sample Report**

```
****** File system cache statistics, severity 1
                        hit %   attempt rate
Directory index cache   95%     3
Directory cache         98%     2
File header cache       85%     4
File id cache           97%     1
Extent cache            87%     1
Bitmap cache            30%     7
```

This event record contains information on how effective the XQP caches are on your system. The first column is the hit percentage and the second column is the attempt rate per second (i.e. how often you are trying to locate the data in file cache).

The brief format of this report does not include those entries which have an attempt rate less than that specified by this parameter. The full format includes all entries with a non-zero attempt rate.

The file system cache sizes are controlled by the ACP system parameters. Low hit rates cause performance problems. If the hit rate is not at least 85% to 90%, increase the associated cache. Keep in mind, that a low hit rate on a cache that is being accessed infrequently is not much of a concern, since the access rate is so low.

# Chapter 7
# User-defined Event Monitoring

## Overview

User-defined event monitoring gives you the ability to set up other events to be monitored by OSM. With user-defined event monitoring you can have OSM monitor any application or event that you would like to have monitored on the system. After user-defined monitoring has been established, you can then cause OSM to perform any of the exception detection actions that OSM normally performs upon detection of your user-defined exception events.

For example, you can monitor the system for the existence of a process that is critical to your application. If OSM detects that the process no longer exists, you can have OSM report to you, page you, or invoke a user-defined fix routine you have specified. The user-defined fix routine might restart the critical process and page you so that you can log in and make sure everything is OK.

## User-Defined Event Monitoring Operation

You specify a DCL command, command procedure or batch file to execute when setting up a user-defined event. This procedure notifies OSM of an exception event through an ASCII file. After user-defined event monitoring has been established, the OSM monitor process will execute the procedure every specified interval. Keep in mind that the monitor process can only check to see if it should execute the user-defined monitoring command when it wakes up, either during its next scheduled wake up, determined by the OSM parameter MONITOR_WAKE, or if it receives a mail message in one of its mailboxes. Therefore, you are not guaranteed that the user-defined monitoring command will execute exactly when the interval specified by the COMMAND# parameter expires.

The OSM monitor process executes the user-defined event monitoring procedure in the context of a subprocess. When the subprocess terminates, the monitor process will be notified. Upon notification, it checks the OSM$LOCAL directory for the existence of the OSM_COMMAND#.severity file, placed there by the user-defined event monitoring command or command procedure. If it exists, the monitor process will include the ASCII information in its binary event file. If the flags argument is set to 1, it will replace any previous exception event information collected for this event with the new exception event information. If the flags argument is set to 0, the binary event information will be included along with any previous exception event information that exists in the binary event file for this user-defined event.

The monitor process will then determine if the severity of the user-defined event, from the file extension, is greater than or equal to the OSM parameter CRITICAL. If so, the .ACTIVE binary event file is renamed to a .SEND binary event file and the transport process will process it. If the severity is less than the OSM parameter CRITICAL, the exception event information is just placed in the .ACTIVE binary event file as mentioned above.

If multiple files are created by the user-defined event monitoring command or command procedure, all of the files will be processed starting with the lowest severity file with the lowest version number through the highest severity file with the highest version number. Once they have been processed, the file(s) created by the user-defined event command or command procedure are deleted, and depending on the flags argument, included in the .ACTIVE binary event file.

# Setting up User-Defined Event Monitoring

To enable user-defined monitoring, you must do the following:

1) Decide what OSM is to monitor.

   OSM can monitor any event that you can monitor with a DCL command or command procedure. You decide what is important to your system or application and determine which command or commands will output the information needed to determine if there is a problem.

2) Create the command or command procedure to perform the monitoring.

   If you have determined that a single command can establish whether a problem exists, all you need to do is set up the OSM parameter COMMAND# to reference the single command, as discussed in the next step.

   If you have determined that you must write a procedure to monitor for your exception event, write and test the command procedure and set up the OSM parameter COMMAND# to reference the command procedure.

   In either case, the command or command procedure must have a way of detecting the event for which you are monitoring. Once it detects the event it places an ASCII file in the OSM$LOCAL directory named OSM_COMMAND#.severity. The # is the same number used in the OSM parameter COMMAND#. The OSM monitor process uses this number to tie the COMMAND parameter with the output files created with that parameter. The severity level is a number from 1 through 10 which identifies the severity of the event.

3) Edit one of the OSM parameter files to include the COMMAND# parameter that instructs the OSM monitor process to monitor a user-defined event.

   Next the desired OSM parameter file must be edited. Edit the node-specific parameter file if the event is to be monitored only on that node, or edit the cluster-wide parameter file if the event is to be monitored on all nodes in the VMScluster. Enter a line in the parameter file that includes the COMMAND# parameter in the form of:

   COMMAND#=seconds,flag,command

   The # is a number in the range of 1 through 1000. Specify a unique number for each user-defined event that you want monitored. This number is also be used by the user-defined monitoring command or command procedure as part of the name of the ASCII file it creates when it detects an event. If you want paging or automated fixes, this number is used in the associated parameter name. Basically, this number ties all aspects of OSM related to user-defined monitoring together.

   The seconds argument determines how often the monitor process will execute the DCL command (or procedure). Each time OSM wakes up, it checks to see if the specified number of seconds has elapsed since the last time the command was executed. If it has been long enough, then the command will be executed again and OSM will check for the existence of its associated ASCII file.

   The flags argument can either be a one (1) or zero (0). It controls the data placed in the binary event file. As mentioned above, the user-defined monitoring command or command procedure will place an ASCII file in the OSM$LOCAL directory when it detects an exception event. If the flags argument is set to 1, OSM will continuously replace the event information in the binary event file with any new event information reported by the user-defined monitoring command or command procedure. If the flags argument is set to 0, OSM will place all information in the binary event file that was reported by the user-defined event monitoring command or command procedure.

   The command argument is the command, command procedure or batch procedure that is executed by the monitor process every specified number of seconds. If this references an DCL command procedure, the command is @file-name.

4) In some cases you may wish to be paged, or have OSM invoke a fix when it detects the user-defined monitor events.

   If you want a page or a fix to be invoked, edit the desired parameter file with the respective OSM parameter, either COMMAND#_P or COMMAND#_F. Make sure that the # is the same as what was assigned in the OSM parameter COMMAND# since it is the only thing that will tie paging or fixes to the user-defined event. For more information on OSM paging, refer to *Chapter 10: OSM Paging Support* in this manual. For more information about automated fixes, refer to *Chapter 11: Automated Fixes* in this manual.

5) The last step is to stop and restart the OSM monitor process to have it recognize the parameter changes. This is accomplished with the OSM commands STOP MONITOR and START MONITOR.

### User-Defined Event Monitoring Example

The following example illustrates how to set up the OSM parameter COMMAND# to execute the show error command. In this example OSM will notify you if the DCL command SHOW ERROR detects any errors. The following would be placed in the appropriate parameter file:

```
COMMAND1 = 1,1, SHOW ERROR/OUTPUT = OSM$LOCAL:OSM_COMMAND1.1
```

This parameter will cause the SHOW ERROR command to be executed every time OSM wakes up, but no more frequently than once each second. The output from that command will be put in a file named OSM_COMMAND1.1 in the OSM$LOCAL directory. The file type of .1 specifies a severity of 1. Since the flag is 1, only the last SHOW ERROR output will be in the OSM data. This means that every message received from the OSM monitor process will include the latest SHOW ERROR output.

## User-Defined Monitoring Template Files

There are several template command procedures that you may use as they are, or modify them to suit your needs. They are in the main OSM directory and have the following file naming convention:

procedure-name_UDM.TEMPLATE

The procedure-name describes what the command procedure will monitor. UDM stands for User-Defined Monitoring, and they all have a .TEMPLATE file extension.

To use them, copy the desired file to a .COM file, edit it if you need to, and edit the appropriate parameter file referencing them.

# Monitoring Disk Controllers and Terminal Servers

As part of user-defined monitoring, OSM allows you to set up routines that OSM will execute to interrogate remote intelligent controllers, such as terminal servers, HSCs and DSSI controllers. Since actual internal controller errors may not be reported to the ERRFMT process, thus not put in the errorlog, the operating system or OSM cannot, by default, report those to you. For example, if an HSC had an internal memory error, the only way you would know about it is if you accessed the HSC and issued the command to display memory error information. This functionality of OSM requires the use of DECnet phase IV.

This feature provides you with the ability to have OSM automatically monitor and report on events that happen on the controllers that you have set up to monitor.

OSM includes are two UDM template files which you can use to set up monitoring of terminal servers and intelligent disk controllers. They are:

### DS100_DS200_MONITOR_UDM.TEMPLATE

This command procedure monitors DECserver 100 and DECserver 200 terminal servers for error events using the SHOW SERVER STATUS and SHOW SERVER COUNTER commands on the server.

### DS90LPLUS_MONITOR_UDM.TEMPLATE

This command procedure monitors DECserver 90L Plus terminal servers for error events using the SHOW SERVER command on the server.

### DSSI_MONITOR_UDM.TEMPLATE

This command procedure monitors DSSI disks (RF30, RF70, RF71, etc.) for exception events using the STATUS CONFIG, STATUS DATALINK and STATUS LOGS commands.

# User-Defined Event Monitoring Commands

## CONNECT

The CONNECT command allows a user to have OSM connect to a storage controller; such as a DSSI disk, HSC, a terminal server, or a printer server, and execute commands on that controller or server to be saved in a file. This may be used by OSM's user-defined event monitoring feature to monitor these devices for specific events. These commands are necessary because the commands that are provided by OpenVMS with similar functions all require a terminal to execute them interactively. The CONNECT command within OSM is designed to execute in batch mode and, when communicating with a terminal or printer server, requires its input to be from a file.

### Format

**CONNECT Controller-server-name**

### Parameter

**Controller-server-name**

Specifies the name of the terminal server or storage controller to be monitored. This parameter is required.

If the controller-server-name is a terminal server, the service circuit and hardware address must be defined in the NCP database. If it is not, then the following error message will be displayed.

```
%SYSTEM-F-INV_HW_ID, invalid hardware identifier (reserved for future use)
```

### Qualifiers

**/INITIAL_WRITE**

Causes this command to send the first command from the input to the storage controller before waiting for the storage controller to prompt for further input. This qualifier is valid only when using the DUP protocol. It is needed when communicating with an HSC50 to get the first control C character written to the storage controller which causes the program on the HSC50 to request further input.

**/INPUT=SYS$INPUT**

Specifies the name of the input file from which all commands will be retrieved to send to the server or storage controller. If using the MOP protocol, this must point to a file, not a terminal.

**/OUTPUT=SYS$OUTPUT**

Specifies the name of the output file to which all output from the server or storage controller will be written.

**/PROTOCOL=DUP**

Specifies the protocol to be used. Valid options are either DUP or MOP. The DUP protocol is used to communicate with storage controllers and the MOP protocol is used to communicate with terminal and printer servers. The default is DUP. The DUP protocol provides similar functionality as the SET HOST/DUP or SET HOST/HSC commands. The MOP protocol provides similar functionality as the NCP CONNECT command.

**/SERVER=MSCP$DUP**

Specifies the name of the communications server when using the DUP protocol to communicate with storage controllers. This is the same as the /SERVER qualifier with the SET HOST/DUP command.

**/TASK=task-name**

Specifies the name of the program to execute on the storage controller. This qualifier is required when using the DUP protocol. This is the same as the /TASK qualifier with the SET HOST/DUP command. The valid options are based on the storage controller to which OSM is connecting.

# User-Defined Event Monitoring Parameters

## COMMAND#=seconds,flags,command

This parameter allows a user specified command to be executed to extend the monitoring capabilities of OSM. The number sign (#) is unique for each user-defined event you would like to have OSM monitor. The number can be in the range of 1-1000.

**Parameter Values**

- Seconds – number of seconds that must elapse before the command is reexecuted.
- Flags –a binary value which determines if the user-defined output is overwritten in the binary event file. A value of 1 causes new information to replace previous information from the same event in the binary event file. A value of 0 causes each user-defined output for this event to be a separate record in the binary event file.
- Command –may be any valid DCL command including @file-name.

Related parameters: COMMAND#_F and COMMAND#_P. See *Chapter 11: Automated Fixes* and *Chapter 10: OSM Paging Support* in this manual for more information on these parameters.

# Chapter 8
# Network Event Monitoring

## Overview

OSM has the ability to monitor remote nodes on the network. It does this by having the OSM monitor process on one node communicate with the monitor process on a remote node. This verifies that the remote system is up and running, that the network between the two systems is functional and that OSM is busy monitoring that system.

In many environments, including those with a distributed database, it is critical that time be synchronized among multiple nodes. OSM can monitor time on a remote system as compared with the time on the local system. It can report as an event when the time difference between the two nodes is greater than a user defined value. OSM is also able to set the time of the local node to that of a remote node to keep the time synchronized.

OSM for OpenVMS supports the use of DECnet phase IV, DECnet Plus and TCP/IP. It has been tested with Multinet, TCPware and HP/Digital TCP/IP Services for OpenVMS, formerly known as UCX.

## Network Event Monitoring Operation

To have OSM monitor either the network, or the time of remote nodes, it must be provided some information. The first step is to set up the network communication on the local and remote nodes. Then OSM must be informed of which nodes to monitor.

### Setting up Network Event Monitoring

When executing in a network environment, the OSM monitor process may need to communicate via a network to the OSM monitor process on other nodes of the network. This is done when the OSM parameters are set up to:

1. monitor operation of remote nodes of a network
2. monitor time differences on multiple nodes of a network
3. synchronize time on multiple nodes of a network
4. synchronize SYSUAF files in a network

Several features exist in OSM to allow multiple OSM environments in one network and to keep the communication secure from other nodes.

There is one parameter and one command that exist to allow the OSM monitor process to communicate via either DECnet or TCP/IP to the OSM monitor process on other nodes. There

is another parameter that is specific to DECnet and one specific to TCP/IP to complete the setup process. The DECNET_OBJECT parameter specifies the name of the object that will be registered with DECnet when OSM is running. The TCPIP_PORT parameter specifies the number of the TCP/IP port on which OSM will listen for requests from OSM on other nodes. The NETWORK_ID parameter specifies a numeric value that must be agreed upon by the different OSM monitor processes that are communicating. The SET PASSWORD command within OSM sets a password which is passed between the OSM processes to verify that those systems are allowed to communicate with each other. Both parameters and the password must be the same on all nodes that are communicating with each other.

OSM may use DECnet to communicate with some nodes of the network and TCP/IP to communicate with other nodes. To set up OSM to communicate in a network environment, the following steps must be taken. These steps should be done on each node of the network that is to participate in one of the above functions within OSM.

1. Determine a name for the DECnet object to be registered on all nodes in the network or the number of the TCP/IP port to be used by OSM. Enter the name as the DECNET_OBJECT parameter on each node. Enter the TCP/IP port number in the TCPIP_PORT parameter on each node.
2. Select a value between 1 and 4000000000 to be used as a unique identifier for all nodes in the network. Enter this value for the NETWORK_ID parameter on each node.
3. Specify either DECNET or TCPIP for the DEFAULT_NETWORK parameter if the default supplied for your operating system is not the one you wish to use.
4. Execute the READ PARAMETERS command. This loads the new parameters into memory for the OSM control image.
5. Execute the OSM command SET PASSWORD. This will prompt for a password that will be saved and used to verify that each node attempting to communicate with the OSM monitor process is allowed.
6. Stop and restart the OSM monitor process. This is done with the commands STOP MONITOR and START MONITOR.
7. Proper set up may be verified with the SHOW TIME/NODE=xxx command where xxx is the name of the node to which communication is being verified. This command should display the time of the specified node. Be sure to specify either the /DECNET or /TCPIP qualifiers with this command if not using the network specified by the DEFAULT_NETWORK parameter.

If the OSM command, SHOW TIME, does not display the current time of the specified node, then an error message will be displayed. The following is a list of the more commonly seen error messages and their causes. Once the cause has been corrected, re-execute the SHOW TIME command in OSM to verify that everything is functioning properly.

1. `%SYSTEM-F-INSFARG, insufficient call arguments`
   This message indicates that the default network has not been defined. Add the DEFAULT_NETWORK parameter to one of the parameter files for this node. Valid values include DECNET and TCPIP. After adding this parameter, restart the monitor process.

2. `%SYSTEM-F-NOSUCHNODE, remote node is unknown` (DECnet)
   This message indicates that the node specified in the SHOW TIME command is not known in the DECnet database on the local node. This is corrected by defining the name and address of the remote node using NCP. Define the node with the command MCR NCP DEFINE NODE xxx ADDRESS 1.2 where xxx is the name of the remote node and 1.2 is the DECnet address of the remote node. Then execute the command MCR NCP SET NODE xxx ALL, again using the name of the remote node for xxx.

3. `%SYSTEM-F-NOSUCHNODE, remote node is unknown` (DECnet)
   `%RMS-E-FNF, record not found` (TCP/IP)
   `%SYSTEM-W-ENDOFFILE, end of file` (TCP/IP)
   `%SYSTEM-W-ENDOFFILE, end of file` (Multinet)
   `%SYSTEM-F-NOSUCHNODE, remote node is unknown` (Multinet)
   These messages indicate that the node specified in the SHOW TIME command is not known in the host database on the local node. This is corrected by defining the name and address of the remote node in the hosts file. For TCP/IP Services, this is done with the command TCPIP SET HOST xxx /ADDRESS=1.2.3.4 where xxx is the name of the remote node and 1.2.3.4 is the IP address of the remote node. You may have the host defined in a name server being referenced by this system.

4. `%SYSTEM-F-UNREACHABLE, remote node is not currently reachable` (DECnet)
   `%SYSTEM-F-TIMEOUT, device timeout` (TCP/IP)
   This message indicates that the network is unable to access the remote node. This means that either there is a problem with the network between the local and remote node, or that the remote node is not currently up or the network is not running on that node.

5. `%SYSTEM-F-NOSUCHOBJ, network object is unknown at remote node` (DECnet)
   `%SYSTEM-F-LINKEXIT, network partner exited` (DECnet)
   `%SYSTEM-F-REJECT, connect to network object rejected` (TCP/IP)
   `%WSAECONNABORTED, Connection refused` (TCP/IP)
   These messages indicate that OSM was able to initiate communication with the remote node, but the network was unable to attach to the specified DECnet object or TCP/IP port on that node. This indicates that either OSM is not running on the remote node, or that OSM did not register the specified network object or listen on the specified port. Verify that the OSM parameter DECNET_OBJECT or TCPIP_PORT is the same on both nodes and that the OSM monitor process is using the current parameters. This may be done with the command OSM SHOW PARAMETERS /MONITOR. Then verify that the OSM monitor process is running on the remote node using the OSM SHOW STATUS command.

6. `%SYSTEM-F-LINKABORT, network partner aborted logical link` (DECnet)
   `%SYSTEM-F-LINKDISCON, network partner disconnected logical link` (TCP/IP)
   `%SYSTEM-F-NOLISTENER, specified remote system process not listening` (TCP/IP)
   This message indicates that the network is functioning between the local and remote node, but that the OSM password or network id is not the same on both nodes. Verify that the NETWORK_ID parameter is the same on both nodes. Then execute the READ PARAMETERS command and the SET PASSWORD command on the newly set up node. Reenter the password when prompted. Stop and restart the OSM monitor process after changing the network id and password.

7. `%SYSTEM-E-UNSUPPORTED, unsupported operation or function`
   This message indicates that TCP/IP, or DECnet was not detected as being installed when OSM was installed. If OSM was installed before the network software, then relink the OSM application and restart it. This is done with by executing the command @OSM$:OSM_ADDNODE. This procedure will ask a couple of questions, relink the OSM images and restart OSM using the new images.

8. `%C-F-NO_TCP, Cannot find TCP routine or no UCX$IPC_SHR` (TCP/IP)
   This message indicates that TCP/IP is not detected as being installed and running when the OSM command was executed. If TCP/IP is installed, then start it. If TCP/IP is not installed, then OSM will be unable to use TCP/IP for network monitoring.

# Network Event Monitoring Commands

## SET PASSWORD

This command may be used to set the password to be associated with the NETWORK_ID for securing the network communication. This command should be executed after changing the NETWORK_ID parameter. If a password has not been set, then all network communication by OSM will not work. The OSM monitor process must be stopped and restarted before the new password will take effect.

This command prompts the user for the new password. The password entered will not be echoed on the terminal. The user will then be prompted for verification of the password entered. If the new password and the verification of the password do not match, OSM will return the following status.

```
%SYSTEM-F-OPINCOMPL, operation is incomplete
```

### Format

**SET PASSWORD**

## SET TIME

This command communicates with the OSM monitor process on the specified node via the network to display and set the time on the local node to match the time of the remote node. It also displays the time of the local node before changing it. OpenVMS requires the user to have LOG_IO privilege to execute this command. This command also requires that the OSM monitor be running on the remote node.

**Format**

**SET TIME**

**Qualifiers**

**/DECNET**

Specifies that this command should use DECnet as the communication path between the local node and the remote node. If neither DECNET nor TCPIP are specified, then the network used is controlled by the DEFAULT_NETWORK parameter.

**/NODE=node-name**

Specifies the name or address of the node for which the time should be retrieved to display and set the time on the local node. If specifying a TCP/IP address, enclose the full address in quotes.

**/TCPIP**

Specifies that this command should use TCP/IP as the communication path between the local node and the remote node. If neither DECNET nor TCPIP are specified, then the network used is controlled by the DEFAULT_NETWORK parameter.

## SHOW TIME

This command communicates with the OSM monitor process on the specified node via the network to display, on the local node, the time of the remote node. This command requires that the OSM monitor be running on the remote node.

**Format**

**SHOW TIME**

**Qualifiers**

**/DECNET**

Specifies that this command should use DECnet as the communication path between the local node and the remote node. If neither DECNET nor TCPIP are specified, then the network used is controlled by the DEFAULT_NETWORK parameter.

**/NODE=node-name**

Specifies the name or address of the node for which the current time should be displayed. The default is to display the time of the local node. If specifying a TCP/IP address, enclose the full address in quotes.

**/TCPIP**

Specifies that this command should use TCP/IP as the communication path between the local node and the remote node. If neither DECNET nor TCPIP are specified, then the network used is controlled by the DEFAULT_NETWORK parameter.

# Network Event Monitoring: Parameters

This section describes the parameters related to network event monitoring.

## DECNET_OBJECT=0

When DECnet is to be used for network communication, OSM must register itself as a network object. The name of the object must be the same for all nodes that are to communicate with each other.

The DECNET_OBJECT parameter controls if the OSM monitor process will register itself as a network object with DECnet. If this option is enabled, a message will be added to the OSM log file indicating either that it successfully registered as a DECnet object or that it was unable to register as a DECnet object and gives the reason. Specifying this parameter without any value causes OSM to not attempt to register as a network object. This is the default.

**Parameter Values**

- Seconds – number of seconds that must pass before each unsuccessful attempt to register as a DECnet object. If the number of seconds is zero, then OSM makes only one attempt to register as a DECnet object when the monitor process starts up. If it is non-zero, then OSM will retry to register itself as a network object with that frequency.
- Name – the name of the DECnet object to be registered. This name is not case sensitive and is registered in uppercase.

Setting this parameter to a non-zero value allows OSM to be started before DECnet and still have everything work correctly. It is also possible to bring DECnet down and back up and have OSM monitor the system while DECnet is down, and resume monitoring of the network when DECnet is restarted.

## DEFAULT_NETWORK=DECNET

This parameter specifies whether DECnet or TCP/IP should be used as the default network for commands or parameters that do not specify a network.

**Parameter Values**

- Network – specifies the default network. The valid values are DECNET for DECnet and TCPIP for TCP/IP. The default value is DECnet.

## NETWORK#=severity,seconds,node-name,network

This parameter causes OSM to monitor the status of a remote node in the network. An event is reported each time the remote node is checked but is not available. To be available, the node must be up and running, the OSM monitor process must be running on that node, and the network must be functional between the two nodes. The # in this parameter is a number in the range of 1 to 1000.

**Parameter Values**

- Severity – the severity of the events reported when the node is not available.
- Seconds – the number of seconds that should elapse between each check for the remote node.
- Name – the name of the remote node.
- Network – the network to be used. The valid values are DECNET or TCPIP. If neither value is specified, then the value specified or defaulted for DEFAULT_NETWORK will be used.

## NETWORK_TIME#=severity,seconds,node-name,delta-time,network

This parameter monitors the time on the local node as compared to a remote node on the network. If the time between the two nodes is off by more than the amount specified in the delta-time portion of this parameter, an event is reported. For this feature to work, the node must be up and running, the OSM monitor process must be running on the remote node, and the network must be functional between the two nodes. The # in this parameter is a number in the range of 1 to 1000.

**Parameter Values**

- Severity – the severity of any event reported when the time on the local node is not within the specified range of the remote node.
- Seconds – the number of seconds that should elapse between each check of the time from the remote node.
- Name – the name of the remote node.
- Delta-time – the amount of difference allowed in time between the local and remote nodes specified in OpenVMS delta time format.
- Network – the network to be used. The valid values are DECNET or TCPIP. If neither value is specified, then the value specified or defaulted for DEFAULT_NETWORK will be used.

## NETWORK_TIME_SET#=node-name,network

This parameter allows OSM to set the time on one node based on the time of another node of the network. The node names specified by the NETWORK_TIME_SET parameters are used to check for the current time. If that time is off from the current system by more than the value specified by the TIME_SET parameter, the time will be set on the local node. The # in this parameter is a number in the range of 1 to 1000. If more than one node is listed, they are checked in the order of their number and the first node found will be used to set the current time.

**Parameter Values**

- Name – the name of the remote node.
- Network – the network to be used. The valid values are DECNET or TCPIP. If neither value is specified, then the value specified or defaulted for DEFAULT_NETWORK will be used.

## NETWORK_ASCII=0

Once the OSM monitor process registers itself as a network object in DECnet, or listens on a TCP/IP port, it is possible for any application on a node in the network to send a message to OSM to be included in its reports. This is done in OpenVMS by writing to the object specified by the DECNET_OBJECT parameter or by writing a text message to the TCP/IP port number that is specified by the TCPIP_PORT parameter.

**Parameter Values**

- Severity – the severity to be assigned to the ASCII data received from network connections. The ASCII data received may change the severity by having the first part of the data be OSM=1 where 1 represents the new severity to be associated with that text. This does not affect the severity of any other messages received. Setting this parameter to 0 prevents any messages from being reported, even if the message attempts to set the severity to another level.

**Note:** The NETWORK_ID parameter and the OSM command SET PASSWORD do not affect this option.

## NETWORK_ID=0

This parameter allows securing and separating different environments of OSM on a single network. This parameter must be the same on all nodes that are to communicate with each other via OSM. After this parameter is changed, the OSM command SET PASSWORD command should be executed to set the password to match the password on the other nodes to which OSM will be communicating.

## TCPIP_PORT=0,0

When OSM is to communicate using TCP/IP, it must create a socket on a port to listen for requests from other nodes. This port number must be the same for all nodes that are to communicate with each other.

This parameter controls if the OSM monitor process will listen for connection requests on a TCP/IP port. If the option is not disabled, a message will be added to the OSM log file indicating either that it successfully opened the specified port or that it was unable to do so and gives the status reason. Specifying this parameter without any value causes OSM to not attempt to listen on any TCP/IP port. This is the default.

**Parameter Values**

- Seconds – the number of seconds that must pass before each unsuccessful attempt to listen on the specified port. If the number of seconds is zero, OSM makes only one attempt to listen on the specified port when the monitor process starts up. If it is non-zero, then OSM will retry listening on the port at that frequency.
- Port – the port number on which OSM should listen. This allows multiple environments to exist on the same network and not allow them to accept communication from each other.

Setting this parameter to a non-zero value allows OSM to be started before TCP/IP and still have everything work correctly. It also enables bringing TCP/IP down and back up and have OSM resume operation.

## TIME_SET=300,0 0:0:15

This parameter specifies how frequently OSM will set the time on the local node to match the time on a remote node.

**Parameter Values**

- Seconds – the frequency in seconds that OSM will check the time from a remote node.
- Delta-time – the amount of time by which the local node must be off from the remote node before OSM changes the time on the local node to match the remote node. This is specified in OpenVMS delta time format.

This parameter affects the NETWORK_TIME_SET# parameter.

# Chapter 9
# Exception Event Reporting

## Overview

The OSM transport process handles exception event reporting. This process is responsible for locating, processing and sending the exception information contained in ".SEND" binary event files, invoking any pages that have been generated by the monitor process, and invoking severity level pages. Refer to *Chapter 10: OSM Paging Support* in this manual for more information on OSM paging support. ".SEND" binary event files, containing all data collected since the last data was sent, are created:

- when the monitor process is restarted if there is a ".ACTIVE" binary event file
- when the monitor process detects an exception event with a severity level greater than or equal to the parameter CRITICAL
- when an errorlog entry threshold has been exceeded
- once each night if the monitor process has detected and saved non-critical exception event data
- by the following OSM commands:

  OSM SEND ALL
  OSM SEND TUNING_INFORMATION /NOW
  OSM SEND CONTACT_INFORMATION /NOW
  OSM SEND REQUEST /NOW

- if the monitor process determines a pager must be paged
- when the transport process on another node sends a ".SEND" file via the network

Once the .SEND binary event files exist, the transport process must discover, process and send the information to the designated location. The transport process searches for the existence of .SEND binary event files each time it wakes up. The transport process wakeup interval is determined by the OSM parameter TRANSPORT_WAKE, which is measured in seconds. The monitor process may wake up the transport process whenever it renames a ".ACTIVE" file to ".SEND".

The monitor process also checks once each day for .ACTIVE binary event files containing non-critical information. If it locates an .ACTIVE binary event file, it will rename it to a .SEND binary event file. During the following transport wake up interval, the transport process will send that .SEND binary event file, as well as any others it finds at one time to the designated destinations. The time of this once-a-day check is determined by the OSM parameter NIGHT.

This parameter defines a start time and a stop time that is the range of time during which the transport process performs this task.

The transport process only needs to run on one node in a VMScluster environment. This transport process will discover, process and send all of the .SEND binary event files it finds, regardless of which node collected the data. When the transport process discovers a .SEND binary event file, it will look at the OSM parameter OUTPUT to determine how to proceed.

# Controlling Output in ASCII Files

This section pertains to ASCII files generated by the OSM transport process. When the OSM transport process detects a .SEND file, it processes it as directed by the OSM parameter OUTPUT. If OUTPUT includes MAIL, FILE or PRINTER in its keyword list, the transport process creates an ASCII file. With parameters, you can control the amount of detail in the file and whether you want to include in the file exception event information that OSM has already reported. This section describes how to set up OSM to get the level of detail you want in your ASCII files.

## Controlling Repeated Exception Events in ASCII Files

Once exceeding the exception event threshold, certain events remain exception events for long periods of time. For example, if free space on a monitored disk exceeds the threshold you have established, it becomes an exception event. If nothing is done to reclaim a sufficient amount of disk space, it remains an exception event.

Depending on your environment, you may want to be notified each time OSM checks and finds a disk that exceeds the threshold. This may generate a lot of exception event messages. You may only want to be notified once a week or once a month of the existing condition. With OSM transport parameters, you can instruct OSM to report on repeated conditions based on your site's needs.

The OSM parameter REPEAT_ALWAYS determines if OSM will report on exception events each time it detects them, even though you have already received an event message for the same event. The OSM parameter REPEAT_DOW will determine if OSM reports on duplicate event information on specified day(s) of the week, and REPEAT_DOM determines if OSM will report on duplicate event information on specified day(s) of the month.

These parameters are valid only when the OSM parameter OUTPUT includes one or more of the keywords MAIL, FILE or PRINTER. If the data being formatted includes only repeated data, and repeated data is not being formatted, the file will not be sent. This eliminates a user receiving a MAIL message with no beneficial information.

## Controlling the Amount of Detail in ASCII Files

The DETAIL parameter will allow you to control the amount of detail you want to see in ASCII files generated by the transport process. It is possible to include a minimal amount of detail in order to keep the messages small. Additional detail may be included to provide more information about the data collected.

Some of the additional information that may be reported includes either a pager notification string that is associated with the event or an automated fix command string that was executed. Security data may be formatted in either a brief or full format.

The amount of detail included is controlled by the TEXT_FORMAT and DETAIL parameters. For a more complete description of these parameters and their associated keywords, refer to the *Transport Process Reporting Parameters* section in this chapter.

ERRORLOG data that was reported to OSM by the ERRFMT process must be formatted before it may be included in a report. This may be done either by OSM, the ANALYZE/ERROR, ANALYZE/ERROR/ELV or the DIAGNOSE command. Each of these methods report different amounts of detail about the reported events. If done with an external command, another parameter, ANALYZE/ERROR, ELV or DIAGNOSE, controls how much information is reported for errorlog entries included in the file. These parameters accept any qualifiers that are valid for the related OpenVMS command. If the data is formatted by OSM, the amount of detail provided is controlled by the DETAIL_ERRORLOG parameter.

# Output Groups

With only one MAIL parameter, all email messages sent by OSM are sent to the same group of people. Output groups extend the flexibility of reporting by allowing multiple settings for most of the parameters that control reporting by the transport process. Each output group may be configured to report on specific events based on:

- Severity of the event
- Reason the file was sent; critical data, nightly processing, pager entry, etc.
- Category of the event; security, performance, hardware errors, etc.
- Account number of the originating system

Each output group is configured with its own set of parameters to control the types of output that are enabled and the related parameters. For example, one output group may send only the security alarms to the security manager via email, while a different output group may send everything except the security alarms to the system manager via email.

Each of the parameters listed in the next section belong to output groups. By default, everything is placed in group 0. When multiple parameter files are used, the default of group 0 is applied to each file. To place parameters in a different group, include the parameter OUTPUT_GROUP. The following example places parameters in two different groups.

```
OUTPUT_GROUP=0
OUTPUT=MAIL
CATEGORY=ALL
TEXT_FORMAT=BRIEF
MAIL=@osm_mail.dis
!
OUTPUT_GROUP=1
OUTPUT=MAIL,PAGER
CATEGORY=SECURITY
TEXT_FORMAT=FULL
MAIL=@osm_security.dis
```

Output group 0 sends all messages, in brief format, via email to the users included in osm_mail.dis. Output group 1 sends only the security messages, in full format, via email to the users in osm_security.dis and invokes any pages that have been generated related to those security messages.

The first time each output group is referenced in a parameter file, it inherits the settings from another group. By default, they are inherited from group 0. Specifying two groups to the OUTPUT_GROUP parameter causes the settings of the second group specified to be used as the initial values for the first group. For example:

```
OUTPUT_GROUP=3,1
```

This causes output group 3 to be created using the current values from group 1 for its initial values. Output group 0 acquires its defaults from the default values for each of the parameters.

The formatting of data by the SUMMARIZE and SUMMARIZE /ERRORLOG commands is controlled by output group 0. Error formatting parameters in output group 0 are used by the monitor process when formatting errolog information. The OPERATOR parameter in output group 0 is used by any process when sending OPCOM messages. If this formatting is to be different than that sent to ASCII formatted output done by the transport process, set the OUTPUT parameter for group 0 to nothing and use a different output group for all formatted output.

# Transport Process Reporting Parameters

The next section of parameters is for use by the OSM transport process. Therefore, when any of these parameters are modified, you must stop and restart the transport process before the changes will take effect.

## ACTIVE=0:00-0:00,Sun,Mon,Tue,Wed,Thu,Fri,Sat

This parameter defines when an output group is active based on the time of day and the day of week. On days or times that an output group is not active; OSM treats the group as though it did not exist. This may be used to change or temporarily disable normal reporting during a regularly scheduled weekly outage.

**Parameter Values**

- The first value in the value list is a start-stop time range based on a 24-hour format.
- The rest of the values define which days the output group is active. The days may be entered as the day of week, with abbreviations as short as three characters being accepted, or as a number of 1 through 7 with 1=Sunday, 2=Monday, ... 7=Saturday.

The default is for each output group to be active through the week.

## ANALYZE/ERROR=

This transport process parameter determines what selected binary error log information is reported to the specified parties when the ERRORLOG_FORMAT parameter is set to ANALYZE. For those outputs that specify ASCII data, such as MAIL, FILE or PRINTER, this binary information must be translated. The OSM transport may do this by issuing the DCL command ANALYZE/ERROR on the collected data. This command has many qualifiers that control the selection and the output format of the ASCII data. This parameter allows you to specify the qualifier(s) to this command controlling the amount of ASCII data that will be output or restrict types of errors that to be included. This parameter accepts any valid qualifiers to the DCL command ANALYZE/ERROR. The default value for this parameter on a VAX is /SUMMARY=(DEVICE, MEMORY, VOLUME, ENTRY)/NOFULL. Since this command does not support the /SUMMARY qualifier on OpenVMS Alpha or OpenVMS Integrity Servers, OSM defaults this parameter to a null string.

## BINARY=file-spec

This transport process parameter specifies the name of the binary event file. This parameter is valid only if the OUTPUT parameter includes the BINARY keyword. Any portion of the file-spec which is omitted from the parameter will default to the name of the input binary .SEND file.

This parameter, along with the BINARY keyword on the OUTPUT parameter, is normally used to make a copy of the binary data to be summarized later with the SUMMARIZE command within OSM. Another common use of this parameter is to send the binary data across the network to be processed by OSM on another node. In both cases, it is normal to specify the device and directory name along with a semicolon. The semicolon is specified to cause OSM to create a new version of the file to be created instead of using the version number of the input file. In the case of sending the data across the network, the node name would also be specified. The file name and extension are normally omitted and therefore defaulted to the name and type of the input file.

Note that the logical name SYS$LOGIN is invalid when you specify a node name in the file specification since OpenVMS does not define this for network processes. An example of this parameter would be:

```
BINARY=VAXC::DUA0:[OSM]
```

This parameter should include both device and directory specification if the output is to be on another node of the network. If using a logical name, RMS will treat the logical name as being a device on the remote node and attempt to fill in the directory specification on the local system. For example:

```
BINARY=NODEA::OSM$:
        should be
BINARY=NODEA::DUA0:[OSM]
```

## CATEGORY=ALL

This transport process parameter affects the type of events included in the output. This allows providing only device errors to the hardware support group; or removing security data from external software support. This affects all OUTPUT formats. This parameter accepts a comma separated list of the desired options. The default is CATEGORY=ALL. The following table lists the valid keywords.

| Keyword | Description |
|---|---|
| ALL | Specifying this keyword enables all other options listed below. It is possible to specify this keyword followed by the negated version of one of the following options. Example: CATEGORY=ALL, NOSECURITY. |
| CAPACITY | Include the capacity planning data collected by OSM during prime time. |
| HARDWARE | Include device error and other hardware specific events monitored by OSM. |
| PERFORMANCE | Include performance related events; such as logical name hash tables. |
| SECURITY | Include security alarms events reported by OSM. |
| SOFTWARE | Include software related events monitored by OSM. |
| TUNING | Include the data collected by the OSM SEND TUNING command. |
| UNKNOWN | Include any event that is not recognized by the version of the transport process that is processing the events. |
| USER | Include user generated events. These are created with the OSM SEND command. |

## DETAIL=SEVERITY

This transport process parameter affects the level of detail that is included in all ASCII output formats. This includes those created by the FILE, MAIL and PRINTER keywords to the OUTPUT parameter. This parameter may be entered as DETAIL which turns off all file detail, or it may be entered as DETAIL=keyword where keyword is a comma separated list of the desired options. The default is DETAIL=SEVERITY. The following table lists the valid keywords.

| Keyword | Description |
|---|---|
| ALL | Specifying this keyword enables all other options listed below. It is possible to specify this keyword followed by the negated version of one of the following options. Example: DETAIL=ALL, NOFILE. |
| CONTACT | Specifies that site contact information detail be included in OSM ASCII files. This includes the company name, contact name and voice phone number contact information in all ASCII output. Refer to *Chapter 4: OSM Identification Parameters* in this manual. |
| FILE | Specifies that information regarding the original binary file be included in the output. This includes the binary event file creation date and time, previous binary event file creation date and time, value of the HEARTBEAT and CRITICAL parameters from the sending node, date and time the binary event file was originally sent and the date and time the monitor process was started. |
| FIX | Specifies that the command string that was executed by the monitor process to correct an exception event be included in OSM ASCII files and reports. For more information, refer to *Chapter 11: Automated Fixes* in this manual. |
| PAGER | Specifies that pager notification strings that have been generated be included in OSM ASCII files and reports. For more information, refer to *Chapter 10: OSM Paging Support* in this manual. |
| REMOVE_TAB | Replaces tab characters with blanks in formatted text output. This is necessary when OUTPUT includes the MAIL or PRITNER keyword, and the target does not support the tab character. This is common with many cell phones and pagers. |

| Keyword | Description |
|---|---|
| SEVERITY | Specifies that the severity level for each group of events be included on the header for that group of events. For example, if page fault statistics has a severity of 6, the severity number will be displayed on the header line that reports page fault statistics. It will also display an asterisk (*) to the right of the statistics report line of the exception event(s) that caused the binary event file to be sent.<br><br>Since each error log entry is not processed individually, the severity displayed on the errorlog header may be a range. This range is for the range of severities of the error log entries. For example, there might be two error log entries in a binary event file, one with a severity of 3, and another with a severity of 5. The reported severity would be displayed as 3-5 indicating the severity range. |

## DETAIL_ERRORLOG=ALL

This transport process parameter affects the level of detail that is included in all ASCII output formats for errorlog data reported. This includes those created by the FILE, MAIL and PRINTER keywords to the OUTPUT parameter. This parameter may be entered as DETAIL_ERRORLOG which turns off all file detail, or it may be entered as DETAIL=keyword where keyword is a comma separated list of the desired options. The default is DETAIL=ALL. The following table lists the valid keywords.

| Keyword | Description |
|---|---|
| ALL | Specifying this keyword enables all other options listed below. This keyword may be specified, followed by the negated version of one or more of the following keywords. For example: DETAIL_ERRORLOG=ALL,NOTAPE. |
| COUNT | Specifies that a count of all errorlog entries be summarized in the report. This keyword is only used when ERRORLOG_FORMAT=OSM is either specified or defaulted. |
| DISK | Specifies that information relating to disk errors be included in the report. |
| ERRORS | Specifies that a summary of all reported errors be included in the report on a device-by-device basis. This keyword is only used when ERRORLOG_FORMAT=OSM is either specified or defaulted. |
| OTHER | Specifies that information relating to errors other than disks or tapes should be included in the report. |
| TAPE | Specifies that information relating to tape errors be included in the report. |
| TIME | Specifies that the date and time of the earliest and latest device errors be included in the report. This keyword is only used when ERRORLOG_FORMAT=OSM is either specified or defaulted. |

## DIAGNOSE=/SUMMARY

This transport process parameter determines what selected binary error log information is reported to the specified parties when the ERRORLOG_FORMAT parameter is set to DIAGNOSE. For those outputs that specify ASCII data, such as MAIL, FILE or PRINTER, this binary information must be translated. The OSM transport process may do this by issuing the DCL command DIAGNOSE on the collected binary data. This command has many qualifiers that control the selection and the output format of the ASCII data. This parameter allows you to specify the qualifier(s) to this command controlling the amount of ASCII data that will be output or restrict types of errors that you want to include. This parameter accepts any valid qualifiers to the DCL command DIAGNOSE. The default value for this parameter is /SUMMARY.

## DISABLED=

This parameter disables an output group based on absolute time. During this time, the output group is treated as though it did not exist. This may be used to change the email addresses

being used for notification while someone is on vacation. This parameter accepts two absolute time values. If the current time is between the two-time values, the output group is disabled. For example: DISABLED=1-MAY-2009,15-MAY-2009 8:00 causes the output group to be disabled from midnight, May 1, 2009 until 8 am, May 15, 2009. Both the ENABLED and DISABLED parameters may be specified. The default is for each output group to be enabled at all times.

## ELV=/SUMMARY

This transport process parameter determines what selected binary error log information is reported to the specified parties when the ERRORLOG_FORMAT parameter is set to ELV. For those outputs that specify ASCII data, such as MAIL, FILE or PRINTER, this binary information must be translated. The OSM transport may do this by issuing the DCL command ANALYZE/ERROR/ELV TRANSLATE on the collected data. This command has many qualifiers that control the selection and the output format of the ASCII data. This parameter allows you to specify the qualifier(s) to this command controlling the amount of ASCII data that will be output or restrict types of errors that to be included. This parameter accepts any valid qualifiers to the DCL command ANALYZE/ERROR/ELV TRANSLATE. The default value for this parameter is /SUMMARY.

## ENABLED=

This parameter enables an output group based on absolute time. During other times, the output group is treated as though it did not exist. This may be used to change the email addresses being used for notification while someone is on vacation. This parameter accepts two absolute time values. If the current time is outside the two-time values, the output group is disabled. For example: ENABLED=1-MAY-2009,15-MAY-2009 8:00 causes the output group to be disabled before midnight, May 1, 2009 and after 8 am, May 15, 2009. Both the ENABLED and DISABLED parameters may be specified. The default is for each output group to be enabled at all times.

## ERRORLOG_FORMAT=OSM

This transport process parameter controls how errorlog information is formatted in reports. It is possible for OSM to format the data or to use one of the OpenVMS supplied utilities to format the data.

| Keyword | Description |
|---|---|
| ANALYZE/ERROR | Specifies that OSM should use the ANALYZE/ERROR command to format all errorlog data. Related parameters: ANALYZE/ERROR. |
| DIAGNOSE | Specifies that OSM should use the DIAGNOSE command to format all errorlog data. Related parameters: DIAGNOSE. |
| ELV | Specifies that OSM should use the ANALYZE/ERROR/ELV TRANSLATE command to format all errorlog data. Related parameters: ELV |
| OSM | Specifies that OSM should format the errorlog data itself. Related parameters: DETAIL_ERRORLOG. |

## ERRORLOG_TRANSLATE=MONITOR

This transport process parameter controls whether the OSM monitor or transport process does the translation of binary errorlog information into an ASCII format. When multiple nodes are providing the information and may have different formats of errorlog information, the monitor process may do the translation to make sure it is translated correctly. If all nodes are using the same errorlog format, then the transport process can do this translation correctly.

## ERRORLOG=OSM$TPG:%04d%02d%02d%02d.DAT

If the OSM parameter OUTPUT includes the keyword ERRORLOG (OUTPUT=ERRORLOG), this transport process parameter specifies the name of the binary error log file where OSM stores the errorlog information. The parameter value can be any file name you specify. Typically, you would want to give a unique name for the saved error log file based on the node the information was collected on as well as some time period. To accomplish this, you have four fields that you can instruct OSM to fill in the account number, from the OSM parameter ACCOUNT, and the month, day and year of when the data was collected. The default value for this parameter, shown above, accomplishes this. The default value causes OSM to store the binary error log data in a file in the OSM$TPG directory. The file name includes 4 characters of the account number (the first %04d), two characters of the month (the next %02d), two characters of the day (the next %02d), and two characters of the year (the last %02d).

If a file already exists with the specified name, OSM appends any new data to the existing file. This usually occurs when data is received from the same node, multiple times in one day or month.

## FILE=OSM$:.OSM_ASCII;

This parameter controls the device and directory location, as well as the file extension, for the ASCII output file if OUTPUT includes the keyword FILE (see OUTPUT). Any portion of the file name not specified defaulted to the name of the binary .SEND file. The default includes the semicolon (;) to force a new version of the file to be created. If this is not specified, and a file already exists with the same or greater version as the input file, OSM creates the output file with a version number of one more than the currently existing file.

It is normal to specify the device, directory name, extension, and a semicolon. The semicolon is specified to cause OSM to create a new version of the file to be created instead of using the version number of the input file. The file name is normally omitted and therefore defaulted to the name of the input file.

## HBOUTPUT=MAIL

This transport process parameter controls where messages are sent that includes only a HEARTBEAT message. Valid keywords are the same as the OUTPUT parameter.

## MAIL=

This transport process parameter accepts valid input to the MAIL utility prompts of TO: or CC:. For example, you may specify a single username, a list of usernames separated by commas, a mail distribution list preceded by the @ character, or if you have UUCP on your system you could specify uucp%"smith@domain.com". If a list of usernames is specified for this parameter, the transport process must be stopped and restarted each time the parameter is changed. If a distribution list is used as a file of usernames, it is not necessary to stop and restart the transport process when the distribution list is changed. This parameter is only valid if the OUTPUT parameter includes the word MAIL (OUTPUT=MAIL). If a mail distribution list is specified, and only a portion of the file name is included, the remaining portions of the file name are defaulted to OSM_MAIL.DIS in the main OSM directory.

The default for the MAIL parameter is a null string. If the OUTPUT parameter includes the keyword MAIL, then this parameter must also be specified.

Since it is possible to include several users to receive mail, it is possible for some users to be able to receive mail while others may not. If OSM is successful in sending mail to at least one user, it will treat that as successfully sending mail and will not retry sending mail to the other usernames. If OSM is unable to send mail to any username, then it will retry sending mail to all users on its next wake up. Those usernames to whom OSM is unable to send mail will be listed in the log file for the OSM transport process.

## OPERATOR=CENTRAL

This parameter specifies which operator terminals will receive OPCOM messages sent by OSM. The default is to send messages to terminals enabled as CENTRAL operators. The valid options are ALL, CARDS, CENTRAL, CLUSTER, DEVICE, DISKS, LICENSE, NETWORK, OPER1-OPER12, PRINT, SECURITY and TAPES. More than one class of operator may be specified by entering a comma separated list of operator classes, e.g. OPERATOR=CENTRAL, OPER5. Specifying the keyword ALL causes all classes of operators to be notified. This may be followed by the negated version of any other operator class to notify all but the specified classes of operators, e.g. OPERATOR=ALL, NOCARDS.

## OUTPUT=MAIL

This transport process parameter is used by OSM to determine where to send the collected exception event data. This parameter and the HBOUTPUT parameter support a comma-separated list of keywords.

| Keyword | Description |
|---|---|
| ALL | Specifying this keyword enables all other options listed below. It is possible to specify this keyword followed by the negated version of one of the following options. Example: OUTPUT=ALL, NOFILE. |
| BINARY | Specifying this keyword causes the transport process to create a copy of the original binary information collected by the OSM monitor process. You would specify this keyword if you want the transport to send binary files over the network to a central location or to save the data for later summarizing.<br>The BINARY parameter is used to specify the name and location of this file. Specifying the BINARY keyword allows you to concentrate all the binary files in one location in a network environment. Once this is done, the OSM_TRANSPORT process at that location can send all the files at once. This may also be used to save a copy of all binary exception event data for future reference, possibly for the purpose of creating a monthly summary. Related parameter: BINARY. |
| ERRORLOG | Specifying this keyword causes the transport process to place all error log exception event information in a separate binary errorlog file. This binary file may then be analyzed with the OpenVMS commands ANALYZE/ERROR, ANALYZE/ERROR/ELV or DIAGNOSE. The name of the binary error log file is determined by the OSM transport parameter, ERRORLOG. You specify this keyword if you are sending summary error log data to a centralized location. This way if the summary data was not conclusive, you have the complete information in binary format. Related parameter: ERRORLOG. |
| FILE | Specifying this keyword creates a sequential ASCII output file that may be read by the system manager or printed. You specify this keyword if you want the transport to create ASCII files in a centralized location for review at any time.<br>The name of the new file is controlled by the parameter FILE. The default is OSM$:.OSM_ASCII;. Any portion of the file name not specified by the FILE parameter will be replaced with that portion of the input file name. Related parameters: FILE, DETAIL, ERRORLOG_FORMAT, DETAIL_ERRORLOG, DIAGNOSE, ELV and ANALYZE/ERROR. |

| | |
|---|---|
| MAIL | Specifying this keyword causes the transport process to format the exception event files in ASCII and mail the ASCII text to destination(s) specified by the OSM parameter MAIL. This is the same information used by the FILE keyword but is sent to a user or list of users via the OpenVMS MAIL utility and provides immediate notification to the users if they are logged into the system receiving mail. This is by far the most common keyword specified for the OUTPUT parameter. Related parameters: MAIL, DETAIL, ERRORLOG_FORMAT, DETAIL_ERRORLOG, DIAGNOSE, ELV and ANALYZE/ERROR. |
| OPCOM | Specifying this keyword causes the transport process to format the exception event files in an abbreviated ASCII message and send the message as OPCOM messages to the operators specified by the OSM parameter OPERATOR. Related parameters: OPERATOR, DETAIL, ERRORLOG_FORMAT, DETAIL_ERRORLOG, DIAGNOSE, ELV and ANALYZE/ERROR. |
| PAGER | Specifying this keyword causes the transport process to invoke any generated pages while processing the binary event file(s). For more information on paging and related parameters, refer to *Chapter 10: OSM Paging Support* in this manual. |
| PRINTER | Specifying this keyword causes the transport process to format the exception event files in ASCII and send the ASCII text to a printer. Related parameters: PRINTER, DETAIL, ERRORLOG_FORMAT, DETAIL_ERRORLOG, DIAGNOSE, ELV and ANALYZE/ERROR. |
| SCRIPT | Specifying this keyword causes the transport process to output either a formatted text file or a binary file and then invoke an external command procedure to complete the processing. The command procedure is provided the name of the file as the P1 parameter. This allows customized processing such as sending the file via ftp. Related parameter: SCRIPT. |

## OUTPUT_ACCOUNTS=

This selects data to be reported based on the account number of the originating system. This allows configuring different levels of reporting for development vs. production systems while having the reporting being consolidated on a single server. The value specified may be multiple account numbers separated by commas; or it may include ranges of numbers separated by dash. For example: OUTPUT_ACCOUNTS=1-100,107,110-120 applies to account numbers 1 through 100, 107 and 110 through 120. The largest account number allowed is 10240. Any account numbers larger than this are ignored.

## OUTPUT_REASONS=ALL

This selects files for reporting based on the reason the file was sent. By default, all files are processed. This parameter accepts a comma separated list of options.

| Keyword | Description |
|---|---|
| ALL | Specifying this keyword enables all other options listed below. It is possible to specify this keyword followed by the negated version of one or more of the following options. Example: OUTPUT=ALL, NONIGHT. |
| CONTACT | This keyword selects files for reporting that were sent by the SEND CONTACT command. |
| CRITICAL | This keyword selects files for reporting that were sent due to the existence of events with a severity equal to, or higher than, the CRITICAL parameter. |
| CRITICAL_PAGER | This keyword selects files for reporting that were sent due to the existence of events with a severity equal to, or higher than, the CRITICAL parameter and included a pager notification string. |
| NIGHT | This keyword selects files for reporting that were sent by normal night-time processing. |
| PAGER | This keyword selects files for reporting that were sent due to the existence of an event with a pager notification string. |
| SEND | This keyword selects file for reporting that were sent by the SEND ALL command. |
| RESTART | This keyword selects files for reporting that were sent by the restart of the monitor process. |
| SHUTDOWN | This keyword selects files for reporting that were sent by the shutdown of the monitor process. |
| TUNING | This keyword selects files for reporting that were sent by the SEND TUNING command. |
| THRESHOLD | This keyword selects files for reporting that were sent due to the existence of a device error that exceeded its threshold. |
| UNKNOWN | This keyword selects files for reporting that were sent for an unknown reason. This can occur if a file with the .ACTIVE extension is renamed to have the .SEND extension. |
| USER | This keyword selects files for reporting that were sent by the SEND REQUEST command. |

## OUTPUT_SEVERITY=0

This transport process parameter sets the lowest severity messages that are included in the output. By default, all messages are included. Setting OUTPUT_SEVERITY=2 causes all severity 1 messages to be excluded from the output. This parameter affects all forms output generated by the transport process.

## OUTPUT_SEVERITY_MAX=10

This transport process parameter sets the highest severity messages that are included in the output. By default, all messages are included. Setting OUTPUT_SEVERITY_MAX=5 causes all severity 6 and higher messages to be excluded from the output. This parameter affects all forms output generated by the transport process.

## PRINTER=

This transport process parameter specifies the name of the printer to send which reports should be sent. This requires that the OUTPUT parameter include the keyword PRINTER. The name of any valid OpenVMS print queue may be specified for this parameter.

## REPEAT_ALWAYS=Y

This parameter determines whether OSM will send repeated exception data to ASCII formatted output; these include FILE, MAIL and PRINTER. If this parameter is set to Y, all repeated data will be sent to the specified parties. A value of N causes OSM not to send repeated exception event information, unless it is defined by the parameters REPEAT_DOW and REPEAT_DOM. Setting this parameter to Y causes OSM to ignore the parameters REPEAT_DOW and REPEAT_DOM. This parameter is valid only when the OUTPUT parameter includes either the FILE, MAIL or PRINTER keywords.

## REPEAT_DOM=0

This parameter specifies which days of the month OSM should restart displaying repeated data. Valid values are 1 through 31 for the days of the month. This may be specified as a comma separated list of days such as REPEAT_DOM=1,15 for the first and 15th of each month. This may be disabled with REPEAT_DOM=0 which is the default. This parameter is valid only when the OUTPUT parameter includes the FILE, MAIL or PRINTER keywords.

## REPEAT_DOW=0

This parameter specifies which days of the week OSM should restart displaying repeated data. Valid values are 1 for Sunday through 7 for Saturday. This may be specified as a comma separated list of days such as REPEAT_DOW=1,4 for Sunday and Wednesday. This is disabled with the default of REPEAT_DOW=0. Specifying REPEAT_DOW=1 will cause OSM to restart displaying repeated data for each node once each week starting on Sunday. This parameter is valid only when the OUTPUT parameter includes the FILE, MAIL or PRINTER keywords.

## SCRIPT=

This parameter controls the type of file provided and the name of the procedure executed if OUTPUT includes the keyword SCRIPT (see OUTPUT). This parameter accepts two values. The first value is either BINARY or TEXT. The second value is the name of the command procedure to be executed. The transport creates either the binary or ASCII formatted output file, then executes the specified command procedure. The name of the file is passed to the command procedure as parameter P1. The procedure may process the file as desired. The input file to the procedure should be left unmodified. When TEXT output files are processed, the text file is formatted based on the same parameters the control the formatting of file output. OSM provides OUTPUT_FTP.TEMPLATE as an example procedure that can send a file to a remote system via ftp.

## TEXT_FORMAT=BRIEF

This transport process parameter affects the level of detail that is included in all ASCII output formats. This includes those created by the FILE, MAIL, OPCOM and PRINTER keywords to the OUTPUT parameter. The default is TEXT_FORMAT=BRIEF. The following table lists the valid keywords.

| Keyword | Description |
|---|---|
| BRIEF | Provides the most critical information needed for each event but may be missing some of the details. For security events, this format is the same as that provided by the ANALYZE /AUDIT /BRIEF command. |
| FULL | Provides the most detailed amount of information available for each event. For security events, this format is the same as that provided by the ANALYZE /AUDIT /FULL command. |
| HEADER_ONLY | Provides a count of the number of events of each type without any additional details. |
| PAGER | Provides a very brief message for each event; designed to be brief enough to fit on a cell phone or pager. |
| SUMMARY | Provides a summary of the number of each type of event. For some types of events, a range of values or the most extreme value will be included. For example, for page and swap file events, the peak page and swap file usage is listed. |

## TEXT_MODE=FILE

This parameter determines if a message is sent for each .ACTIVE file processed, or for each event in each file. Due to limitations on the amount of output that can be accepted by some pagers, cell phones and OPCOM, these are usually processed with TEXT_MODE=EVENT. This includes those outputs created by the FILE, MAIL, OPCOM and PRINTER keywords to the OUTPUT parameter. The default is TEXT_MODE=FILE. The following table lists the valid keywords.

| Keyword | Description |
|---|---|
| EVENT | Reports each event separately, causing multiple files or mail or OPCOM messages to be sent. |
| FILE | Reports all text relating to all events in the file as a single message. |

## TEXT_PREFIX=
## TEXT_PREFIX_ADD=

This transport process parameter specifies text to be included at the beginning of any all ASCII output formats. This includes those created by the FILE, MAIL and PRINTER keywords to the OUTPUT parameter. This allows providing additional notes or text at the beginning of each message. TEXT_PREFIX_ADD may be used to add additional lines.

## TEXT_POSTFIX=
## TEXT_POSTFIX_ADD=

This transport process parameter specifies text to be included at the end of any all ASCII output formats. This includes those created by the FILE, MAIL and PRINTER keywords to the OUTPUT parameter. This allows providing additional notes or text at the end of each message. TEXT_POSTFIX_ADD may be used to add additional lines.

# Chapter 10
# OSM Paging Support

## Overview

OSM has the ability to generate a page on a pager, cell phone or PDA that supports access through email based on the detection of specific events or based on any exception event exceeding a specified severity level. There are several parameters and commands that allow you to control how OSM pages, and upon what events OSM will page. The OSM paging mechanism has the following features:

- Repeat Paging
- Multi-level Backup Paging
- Paging History Tracking

To help understand the various components of the OSM pager subsystem, there are two operations that must be defined: generating a page and invoking a page. Generating a page means that OSM has determined that a pager should be paged and it creates a message string that identifies the event. To invoke a page, the generated string is sent via email to contact the pager.

The OSM monitor process can only generate a page. The OSM transport process can both generate and invoke a page. The only way the transport process can generate a page is if it finds an event with a severity level that meets or exceeds the severity specified by the PAGER parameter while it is processing the binary event file. It invokes pages it generates or those generated by the monitor process while processing the binary event files. Both occur only if the parameter OUTPUT includes the keyword PAGER.

# Operation

## Generating and Invoking a Page

There are three scenarios in which OSM generates a page, two of which are when there is only one transport process responsible for notifying users and invoking pages. In these two scenarios, the transport process' OUTPUT parameter has the keyword PAGER specified (OUTPUT=PAGER). This would typically be the case when OSM is set up on local systems to notify local support personnel of events that might occur on those systems. Those two scenarios are as follows:

1) A monitor process running on any node detects an event and the associated paging parameter has been placed in one of the OSM parameter files. For example, if OSM detects a break-in attempt and the OSM parameter AUDIT_BREAKIN_P has an associated notification string in one of the OSM parameter files, the monitor process generates and stores a page in the binary event file. The transport process invokes the page when it processes that binary event file.
2) The monitor process detects events that exceed the specified threshold and records them in the binary event file. A page is generated and invoked by the transport process when it processes the binary event file and discovers events that have a severity level greater than or equal to that specified by the OSM parameter PAGER.

The last scenario only occurs if you have multiple transport processes reporting to a single transport process at a central location. This is accomplished by setting up the multiple transport processes to upload, via the network, the binary files to the transport process at the central location. This may be done by assigning the BINARY or SCRIPT keyword to the OSM parameter OUTPUT (OUTPUT=BINARY or OUTPUT=SCRIPT) and their associated parameters in one of the OSM parameter files on the multiple transports. Binary output supports transmitting the file using DECnet. Script output may transfer the file using FTP, Kermit or other method available on your system.

The transport at the central location then includes the keyword PAGER with the OSM parameter OUTPUT (OUTPUT=PAGER). This is a typical scenario when there are remote systems spread out across the town or country that are being managed by a centralized management team. Whenever an exception event is detected on the remote or local system, the centralized management team would be notified. The last scenario is as follows:

3) The monitor process on any of the remote nodes detects events that exceed the specified threshold and records them in the binary event file. If any of these events have a page associated with them, the monitor process includes that information in the binary event file. The remote transport process uploads the binary event files to the central management location. A page is generated and invoked when the transport process at the central management location starts processing the binary event file and discovers events that have a severity level greater than or equal to the OSM parameter PAGER. The transport process at the central management location will also invoke any page that was requested in the binary event file by the remote monitor process (Similar to scenario 1).

## Paging Notification String

The paging notification string determines who is notified and the backup notification for each page. The notification string consists of:

- Primary and multiple levels of backup notifications separated by the | character.
- Email addresses to be notified separated by commas.

An example of the paging notification string is:

3035551212@vtext.com,3035551200@txt.att.net|7205551208@message@tmomail.net

Each monitored event has an associated OSM event pager parameter that generates a page if the parameter has been placed in one of the OSM parameter files. For every monitored event there is an event pager parameter in the form of:

event-name_P=notification-string

The event-name is the name of the monitored event. For example, the OSM parameter that causes OSM to generate a page when a power fail recovery has occurred is POWER_P=pager-string. All the pager parameters are listed in this chapter. The _P indicates that it is a pager parameter.

As with the MAIL parameter, distribution lists may be included in the notification string. This is done by specifying the name of the file containing the distribution list preceded by the @ character. For example:

@pager.dis|@escalation.dis

**Popular Cell Phone Email Addresses**

Short Message Service (SMS) is a method of sending a text message to a cell phone. Most cell phones are available at their 10-digit phone number @ a service specific host. For example, 3035551212@vtext.com is used to send a text message to a Verizon cell phone with the number 303-555-1212. Popular services in the US include:

- Alltel – message.alltel.com
- AT&T – txt.att.net
- Cingular – cingularme.com
- Cingular (originally w/AT&T) – mmode.com
- Nextel – messaging.nextel.com
- Sprint PCS – messaging.sprintpcs.com
- T-Mobile – tmomail.net
- Verizon – vtext.com
- Virgin Mobile USA – vmobl.com

Verify the email address before relying on getting paged by OSM. For a list of many more providers, go to http://www.notepage.net/smtp.htm.

## Paging for Errorlog Entries

OSM generates a page for errorlog entries only after the number of errorlog entries for a specific device has exceeded the threshold for that device. For more information on errorlog entry thresholds, reference *Chapter 6: Monitoring for Exception Events* in this manual.

## Repeat Paging

OSM has the ability to repeat a page. The page may be repeated a predetermined number of times, or it may go on indefinitely until that page is cleared. Repeat pages are controlled by the OSM parameter PAGER_RETRY.

The PAGER_RETRY parameter specifies the number of times a pager should be tried, and the amount of time between retries. The format of this parameter is:

PAGER_RETRY=n,dd hh:mm:ss

The number of attempts, **n**, can be any positive integer; a value of zero (0) indicates the page should retry indefinitely. The time between pages is specified as an OpenVMS delta time with the format of **dd hh:mm:ss**, where **dd** is the number of days, **hh** is the number of hours, **mm** is the number of minutes, and **ss** is the number of seconds between pages.

Paging retries are performed by the transport process, which wakes up every WAKE_TRANSPORT interval. This means that the smallest granularity of retry paging is determined by the OSM parameter WAKE_TRANSPORT. Therefore, it is recommended that the value of the OSM parameter PAGER_RETRY be a multiple of the value of the OSM parameter WAKE_TRANSPORT.

The only way to clear a page that is retrying indefinitely is with the OSM command CLEAR PAGER. The format of this command is:

$ OSM CLEAR PAGER **id**

Where **id** is the pager identification which can be discovered by issuing the OSM command SHOW PAGER/OUTSTANDING.

The following example illustrates how to display and clear a page:

```
$ OSM SHOW PAGER/ALL
Pager entries which are OUTSTANDING
Entry:  5      21-NOV-1996 15:50 has not been cleared
        ATDT8917307,,,97#,;H

Pager entries which have been CLEARED by primary
No entries found
Pager entries which have been CLEARED by backup
Entry:  3      18-NOV-1996 16:08 cleared after failover at 21-NOV-1996 09:43
        ATDT8917307,,,97#,;H

Pager entries which have EXPIRED
Entry:  4      21-NOV-1996 14:05 expired at 21-NOV-1996 15:37
        ATDT8917307,,,97#,;H
```

This SHOW command display indicates that there is one page outstanding, entry number 5. There were no pages that were cleared by the primary page recipient, and entry number 3 failed over and was cleared by the secondary page recipient. Entry number 4 was not cleared by either the primary or secondary page recipient and expired.

```
$ OSM CLEAR PAGER 5
Entry:  5      21-NOV-1996 15:50 cleared at 21-NOV-1996 15:51
        ATDT8917307,,,97#,;H
```

This is the command that clears a page once it has been generated. In this command we are clearing the page associated with entry number 5.

```
$ OSM SHOW PAGE/ALL
Pager entries which are OUTSTANDING
No entries found

Pager entries which have been CLEARED by primary
Entry:  5      21-NOV-1996 15:50 cleared at 21-NOV-1996 15:51
        ATDT8917307,,,97#,;H

Pager entries which have been CLEARED by backup
Entry:  3      18-NOV-1996 16:08 cleared after failover at 21-NOV-1996 09:43
        ATDT8917307,,,97#,;H

Pager entries which have EXPIRED
Entry:  4      21-NOV-1996 14:05 expired at 21-NOV-1996 15:37
        ATDT8917307,,,97#,;H
```

Notice that when we executed the show command again, the page associated with entry number 5 shows up as cleared by the primary page recipient, and there are no more outstanding pages.

## Backup Paging

After the number of retries specified by the PAGER_RETRY parameter has been exhausted, OSM checks the pager notification string to see if a backup pager string has been defined. This is a second list of email addresses specified after the | character. Multiple levels of notification strings may be included. Setting the PAGER_RETRY parameter to 0 causes the initial notification string to be tried indefinitely. This disables backup paging.

If the string is terminated with the | character, the page will automatically expire after each group of email addresses have been tried the number of times specified by PAGER_RETRY. If the string does not terminate with the | character, then the last set of email addresses will be retried indefinitely

## Paging History

Once a page has been invoked, the OSM transport process keeps track of the page in a file in the OSM$ directory named OSM_INFO.DAT. By default, this file contains paging information for the previous two weeks. This period is determined by the OSM parameter PAGER_DELETE. The format of this parameter is as follows:

PAGER_DELETE=dd hh:mm:ss

The parameter value is a delta time. The dd is the number of days, hh is the number of hours, mm is the number of minutes and ss is the number of seconds which determines how long a page will be kept in the paging history file.

# OSM Paging Commands

There are two OSM commands that are used to view and clear pager requests. These commands, along with their associated parameters, are documented below:

## CLEAR PAGER

This command clears the specified outstanding pager request. Each pager request that is cleared will be displayed after clearing. To find out which pager identification number to specify, issue the OSM command SHOW PAGER/OUTSTANDING

**Format**

**CLEAR PAGER id**

**Parameter**

**id**

Specifies the pager identification number. This parameter is required if the /ALL qualifier is not specified, and has no default value.

**Qualifiers**

**/ALL**

Specifies that all outstanding pager requests should be cleared.

**/LOG**

Specifies that each pager that is cleared be displayed to the user.

## DELETE PAGER

This command allows a user to remove one or more pager entries from the pager history.

**Format**

**DELETE PAGER [id]**

**Parameter**

**id**

Specifies the pager identification number. If this parameter is specified, information about the status of that page is displayed. If this parameter is not specified, all pager requests that match the search criteria defined by the qualifiers are displayed. If the pager identification number is specified, no qualifiers are allowed. This parameter is optional.

**Qualifiers**

**/ALL**

Deletes all pager requests. This includes both active and archived requests. The /OUTSTANDING, /CLEARED, /EXPIRED and /ALL qualifiers are mutually exclusive.

**/BEFORE=time**

Deletes pager requests that were archived by the transport process before the specified time. You may specify an absolute time, a delta time or a combination of both. For more information on OpenVMS time formats, consult your OpenVMS documentation.

**/CLEARED**

This qualifier deletes the pager requests that have been cleared by the CLEAR PAGER command.

**/EXPIRED**

Deletes the pager requests that have expired. Expired page requests have not been responded to during the time that OSM was invoking the page. The OSM parameter PAGER_RETRY controls expiration time. The /OUTSTANDING, /CLEARED, /EXPIRED and /ALL qualifiers are mutually exclusive.

**/OUTSTANDING**

Deletes all pages that are currently active. A page is active if it has not exhausted its retry count and no one has issued the OSM command CLEAR PAGER for that request. The /OUTSTANDING, /CLEARED, /EXPIRED and /ALL qualifiers are mutually exclusive.

**/SINCE=time**

Deletes pager requests that were archived by the transport process on or after the specified time. You may specify an absolute time, a delta time or a combination of both. For more information on OpenVMS time formats, consult your OpenVMS documentation.

## SHOW PAGER

This command allows a user to see currently active and archived pager requests.

**Format**

**SHOW PAGER [id]**

**Parameter**

**id**

Specifies the pager identification number. If this parameter is specified, information about the status of that page is displayed. If this parameter is not specified, all pager requests that match the search criteria defined by the qualifiers are displayed. If the pager identification number is specified, no qualifiers are allowed. This parameter is optional.

**Qualifiers**

**/ALL**

Displays all pager requests. This includes both active and archived requests. The /OUTSTANDING, /CLEARED, /EXPIRED and /ALL qualifiers are mutually exclusive.

**/BEFORE=time**

Displays pager requests that were first seen by the transport process before the specified time. You may specify an absolute time, a delta time or a combination of both. For more information on OpenVMS time formats, consult your OpenVMS documentation.

**/CLEARED**

This qualifier displays the pager requests that have been cleared by the CLEAR PAGER command.

**/EXPIRED**

Displays the pager requests that have expired. Expired page requests have not been responded to during the time that OSM was paging either the primary page recipient or secondary page recipient. The OSM parameter PAGER_RETRY controls expiration time. The /OUTSTANDING, /CLEARED, /EXPIRED and /ALL qualifiers are mutually exclusive.

**/FULL**

Displays the last time the pager request has been tried, and how many times it has been tried for each of the possible destinations. This is in addition to the other information normally displayed.

**Note**: If the retry count in the PAGER_RETRY parameter is changed after a page has been generated, the number of times per individual may not be accurate.

**/OUTSTANDING (default)**

Displays all pages that are currently active. A page is active if it has not exhausted its retry count and no one has issued the OSM command CLEAR PAGER for that request. The /OUTSTANDING, /CLEARED, /EXPIRED and /ALL qualifiers are mutually exclusive.

**/SINCE=time**

Displays pager requests that were first seen by the transport process on or after the specified time. You may specify an absolute time, a delta time or a combination of both. For more information on OpenVMS time formats, consult your OpenVMS documentation.

# OSM Parameters Related to Paging

This section describes the parameters that are not specific to paging, but have some relation to it. They are discussed in more detail in other parts of the manual.

## DETAIL

This parameter affects the amount of information that is sent to any ASCII file. For example, any mail message sent to you by OSM. If you specify the keyword PAGER (DETAIL=PAGER), the generated notification string will be displayed in all ASCII output. The DETAIL parameter is described in more detail in *Chapter 9: Exception Event Reporting* in this manual.

## OUTPUT

This parameter determines where OSM is going to send output information. If you specify the keyword PAGER (OUTPUT=PAGER), the transport process will invoke any generated pages. The OUTPUT parameter is described in more detail in *Chapter 9: Exception Event Reporting* in this manual.

# General Pager Control Parameters

The section describes the parameters that control the operation of the pager subsystem in general. The following parameters control the pager retry and fail over capabilities. All these parameters are OSM transport parameters. If they are changed, the transport process must be stopped and restarted before they take effect. These parameters are valid only if the OUTPUT parameter includes the keyword PAGER.

## PAGER=0

This parameter instructs the transport process to generate and invoke a page upon detection of exception events with a severity level greater than or equal to the specified severity. The first number is the severity level. Setting this to a value of 0 disables this feature. However, pages generated by the monitor process may still be invoked.

## PAGER_DELETE=14 0:0:0

This parameter controls how long OSM will keep a cleared or expired pager entry in the history database before it is deleted. The only parameter list component is the delta time that specifies how long a page is archived. The transport process deletes pager entries when the time they were cleared becomes as old as the delta time specified. This is done once each night and when the transport process is started.

## PAGER_RETRY=1,0 00:00:00

This parameter controls the frequency and number of times OSM will retry a pager. The parameter value list is made up of two components separated by commas.

**Parameter Values**

- The first component is the retry limit
- The second is the delta time for retries.

Each pager entry will be retried the number of times specified unless it is cleared by the OSM command CLEAR PAGER. The minimum number of retries allowed is 1. The delta time specifies the interval of time that must pass before retrying a pager request. After the delta time has expired, the retry will occur during the transport process' next wakeup interval. Be sure to set the TRANSPORT_WAKE parameter as low as the retry frequency on this parameter.

# Exception Event Paging Parameters

In all of the remaining pager parameters in this chapter, a pager-string is specified. This is the notification string used to determine who is notified for primary and backup notifications.

All of these parameters are OSM monitor parameters. If they are changed, the monitor process must be stopped and restarted before they take effect. They can be placed in either the node-specific or cluster-wide parameter file to have the monitor process generate a page when it detects the associated event.

Each of these parameters end with _P. There is an exception event monitoring parameter associated with each of these parameters that does not include the _P. The exception event monitoring parameters are documented in *Chapter 6: Monitoring for Exception Events*. To be effective, the severity level of the associated exception event monitoring parameter must be set to a nonzero value. If there is both a critical and a non-critical severity level for an exception event, the page is generated only when the monitor process detects the exception at the critical level. Therefore, the severity level of the associated exception event monitoring parameter that must be set to a nonzero severity level is the critical parameter that ends with _C.

For example, to have the BUGCHECK_P parameter generate a page, the associated parameter BUGCHECK must have a severity level greater than 0. To have the DEADLOCK_P parameter generate a page, the associated parameter DEADLOCK_C must have a severity level greater than 0.

## BUGCHECK_P=pager-string

This causes a page to be generated and sent to the transport process anytime a bugcheck is reported by the monitor process. These are the bugchecks that cause OpenVMS to go down and reboot. The ERRFMT process reports bugchecks that do not cause the system to go down. The pages for these bugchecks are controlled by the ERRFMT_BUGCHECK_P parameter.

## CAPACITY_P=pager-string

This causes a page to be generated and sent to the transport process anytime capacity-planning information is reported by the monitor process.

## COMMAND#_P=pager-string

This causes a page to be generated and sent to the transport process anytime a user-defined command is executed which creates the output file to be reported by OSM. The # in the above parameter is any valid number in the range of 1 to 1000 and must match the COMMAND# parameter to which the page is to be associated. For more information on user-defined monitoring, see *Chapter 7: User-defined Event Monitoring* in this manual.

## COMPUTABLE_P=pager-string

This causes a page to be generated and sent to the transport process anytime a compute-bound process is reported by the monitor process.

### CPU_P=pager-string

This causes a page to be generated and sent to the transport process anytime the monitor process detects that the CPU spends an excessive amount of time in kernel mode, on the interrupt stack, or performing multiprocessing synchronization.

### DEADLOCK_P=pager-string

This causes a page to be generated and sent to the transport process whenever excessive deadlock searches or detections are reported by the monitor process at the critical level.

### DISK_P=pager-string

This causes a page to be generated and sent to the transport process anytime the monitor process determines that disk usage exceeds what you have defined as critical. This parameter may be set on a disk-by-disk basis by using the OSM_DISK_xxx logical name where xxx is the name of the disk device. Refer to *Chapter 5: Controlling OSM Processes* in this manual for more information on OSM logical names.

### GBLSECTIONS_P=pager-string

This causes a page to be generated and sent to the transport process anytime free global pages or global sections are reported by the monitor process to be at or below the critical level.

### IO_P=pager-string

This causes a page to be generated and sent to the transport process whenever excessive split I/Os or window turns are reported by the monitor process to be at or above the critical level.

### LOGICAL_NAMES_P=pager-string

This causes a page to be generated and sent to the transport process anytime logical name performance statistics are reported by the monitor process.

### NETWORK_ASCII_P=pager-string

This causes a page to be generated and sent to the transport process anytime an ASCII message is received from the network.

### NETWORK#_P=pager-string

This causes a page to be generated and sent to the transport process anytime a remote node is inaccessible. The # in the parameter is any valid number in the range of 1 to 1000 and must match the NETWORK# parameter to which the page is to be associated.

### NETWORK_TIME#_P=pager-string

This causes a page to be generated and sent to the transport process anytime an event is reported due to differences in time between the local node and a remote node. The # in the parameter is any valid number in the range of 1 to 1000 and must match the NETWORK_TIME# parameter to which the page is to be associated.

### OPCOM_ERROR_P=pager-string

This causes a page to be generated and invoked by the transport process when it is unable to send one or more files to all output destinations. If this occurs consecutively the number of times specified by the OPCOM_ERROR parameter, the transport process will send an OPCOM message and generate the page specified by this parameter.

### PAGEFILE_P=pager-string

This causes a page to be generated and sent to the transport process anytime the free page file or swap file space is reported by the monitor process to be at or below the critical level.

### PAGE_FAULTS_P=pager-string

This causes a page to be generated and sent to the transport process anytime excessive page faulting is reported by the monitor process.

### POOL_P=pager-string

This causes a page to be generated and sent to the transport process anytime memory pool usage or pool allocation failures are reported by the monitor process to be at the critical level.

### POWER_P=pager-string

This causes a page to be generated and sent to the transport process anytime a power failure recovery is reported by the monitor process.

### PROCESSES_P=pager-string

This causes a page to be generated and sent to the transport process anytime a shortage of process slots or balance slots is reported by the monitor process.

### TEMPERATURE_P=pager-string

This causes a page to be generated and sent to the transport process anytime a high temperature event is reported by the monitor process.

### XQP_CACHE_P=pager-string

This causes a page to be generated and sent to the transport process anytime the file system cache statistics are reported by the monitor process.

## Error Format Paging Parameters

This section lists parameters that cause a page to be generated based on the receipt of the associated error message from the ERRFMT process. If the related message is seen, then a page is generated and sent to the transport process to be invoked. The following table describes the error format paging parameters.

| Error Format Paging Parameter Names | Description |
|---|---|
| ERRFMT_BUGCHECK_P | Nonfatal Bugcheck Errors |
| ERRFMT_BUS_P | External Bus Errors |
| ERRFMT_CARD_P | Card Reader Errors |
| ERRFMT_DISK_P | Disk Errors |
| ERRFMT_HARD_MEMORY_P | Hard Memory Errors |
| ERRFMT_INFORMATION_P | Informational Errors |
| ERRFMT_LINE_PRINTER_P | Line Printer Errors |
| ERRFMT_MAILBOX_P | Mailbox Errors |
| ERRFMT_MEMORY_P | Memory Errors |
| ERRFMT_MISCELLANEOUS_P | Miscellaneous Device Errors |
| ERRFMT_POWER_P | Power fail Recoveries |
| ERRFMT_REALTIME_P | Real-time Device Errors |
| ERRFMT_SYNCHRONOUS_ COMMUNICATION_P | Synchronous Communication |
| ERRFMT_SYSTEM_P | CPU, Console and Bus Errors |
| ERRFMT_TAPE_P | Tape Errors |
| ERRFMT_TERMINAL_P | Terminal Errors |
| ERRFMT_VOLUME_P | Volume Mounts and Dismounts |
| ERRFMT_WORKSTATION_P | Workstation Errors |

## Audit Server Paging Parameters

These section lists parameters that cause a page to be generated based on the receipt of the associated security alarm message from the AUDIT_SERVER process. If the related message is seen, a page is generated and sent to the transport process to be invoked. The following table describes the audit server paging parameters.

| Audit Server Paging Parameter Names | Description |
|---|---|
| AUDIT_BREAKIN_P | Break-in Attempts |
| AUDIT_BYPASS_P | Use of Bypass Privilege to Access Files |
| AUDIT_INSTALL_P | Use of the Install Utility to Add or Remove a Known Image |
| AUDIT_LOGFAIL_P | Login Failures |
| AUDIT_READALL_P | Use of READALL Privilege to Access Files |
| AUDIT_UAF_P | SYSUAF Modifications Other Than a User Changing Their Own Password Via Loginout or Set Password. |

# Chapter 11
# Automated Fixes

## Overview

OSM has the ability to execute a command, a command procedure or a batch file to fix an event once it determines that an exception event exceeds either the normal or critical threshold. This OSM functionality is called automated fixes.

For all OSM monitored events, you may specify a fix to be initiated when OSM detects that the monitored event exceeds the threshold specified in the event parameter. For all OSM monitored events that also have a critical threshold, you may specify that a different automated fix be applied when the critical threshold has been exceeded.

Keep in mind that setting either the critical or non-critical event monitoring parameters to 0 will disable the associated event. Disabling an event monitoring parameter will also disable the associated fix parameter.

Implementing automated fixes is accomplished by placing the associated automated fix parameters in either the cluster-wide or node-specific OSM parameter file.

The formats of these parameters are:

xxxx_F=flags,command
xxxx_CF=flags,command

The following table describes each of the parameter elements:

| Element | Description |
|---|---|
| xxxx | The event parameter name. For example, if you had a command procedure to invoke when page fault rates became excessive, the event parameter name would be PAGE_FAULTS. The event parameter names are consistent across OSM components. The parameter for determining if there are excessive page faults, the parameter that determines if a pager should be paged for excessive page faults, and the parameter to invoke an automated fix for excessive page faults all begin with the keyword PAGE_FAULTS. |
| _F | Parameters that end with _F are used to specify commands or command procedures to be executed in an attempt to fix non-critical events. (Monitor event parameters that do not end with _C.) |
| _CF | Parameters that end with _CF are used to specify commands or command procedures to be executed in an attempt to fix critical events. (Monitor event parameters that end with _C.) |
| flags | This parameter is a bit mask of flags passed to the LIB$SPAWN routine. Setting this to zero (0) will cause the OSM monitor process to suspend its own execution until the subprocess executing the command completes. Setting this to one (1) causes the OSM monitor process to continue execution in parallel with the subprocess executing the command or command procedure. This is equivalent to the SPAWN/NOWAIT command. For more information on the flags passed to a subprocess using the LIB$SPAWN run time library routine, refer to the *OpenVMS Run Time Library Routines* manual. |
| command | The command is any valid command string including the @ character on OpenVMS which will execute a command procedure. In most cases, a command procedure or batch file will be preferable since you can execute multiple commands within the procedure. |

Some example settings for a fix parameter would be:

```
CPU_F=0,SUBMIT OSM$:HIGH_KERNEL_TIME.COM/PARAMETER=
DISK_CF=0,@OSM$:DISK_PURGE_FIX.COM
```

In this example, the 0 indicates that the OSM monitor process will wait for the DCL command SUBMIT to complete before it continues processing. Notice that since the SUBMIT command was specified, we do not wait for the command procedure to complete before the monitor process can continue. OSM will only wait for the SUBMIT command itself to complete. If you specify @command-file with the value of 0, the monitor process would have waited until the command file was complete.

The SUBMIT OSM$:HIGH_KERNEL_TIME is the fix command to be executed when OSM detects high CPU overhead. In this case, we are executing a command procedure as a batch job.

The /PARAMETER= only is valid for parameters that pass values to the fix subprocess. When OSM passes parameters, it simply appends the parameter it is going to pass to the end of the

command string. If you place the /PARAMETER= qualifier anywhere else on the command line, the fix attempt will fail. It is OK to use the /PARAMETER qualifier with the SUBMIT command when the parameters are also specified on the command line. This may be done anywhere within the command line.

### Automated Fixes for Errorlog Entries

OSM will execute an automated fix command string for errorlog entries once that number has exceeded the threshold for the specific device. For more information on errorlog entry thresholds, reference *Chapter 6: Monitoring for Exception Events* in this manual.

## Enabling an Automated Fix Audit Trail

To generate an audit trail of fixes, the transport process can report commands that have been executed in an attempt to fix a problem. This is done by adding the FIX string to the DETAIL parameter (DETAIL=FIX) in either the cluster-wide or node- specific parameter files. By default, automated fix auditing will not be implemented. Keep in mind that the transport process will only be able to report this once it has processed the binary event log file.

You may also include the FIX keyword to the LOG parameter. This causes the OSM monitor process to log all automated fix commands to its log file. This log file is in the OSM$ directory and is named OSM.LOG.

## Automated Fix Template Files

Included with OSM are a number of automated fix template files. These files were created to demonstrate how to implement an automated fix, and what some of the common fixes are for some of the monitored OSM events.

The template files are found in the directory pointed to by the logical name OSM$:. They have the name of xxxx_FIX.TEMPLATE, where the xxxx is the name of the event to fix.

# Automated Fix Parameters

All of the OSM parameters that are related to automated fixes are OSM monitor parameters. The monitor process is solely responsible for detecting and implementing automated fixes. Therefore, if any of the automated fix parameters are modified in one of the OSM parameter files, the monitor process must be stopped and restarted before those parameters take effect. These parameters may be placed in either the node-specific or cluster-wide parameter file to have the monitor process invoke the specified command upon detection of the associated event. For each of the automated fix parameters to be valid, its associated exception event parameter must include a nonzero severity level. For each of the automated fix parameters ending in _CF, the critical exception event parameter must be set to a nonzero severity level. For additional information on OSM monitored events, see *Chapter 1: Introduction to OSM* or *Chapter 6: Monitoring for Exception Events* in this manual. All the automated fix parameters are listed below:

## ALIGNMENT_FAULTS_F=flags,command

This causes the specified command to be executed anytime the monitor process reports excessive alignment faults.

## BUGCHECK_F=flags,command

This causes the specified command to be executed anytime the monitor process reports a bugcheck. These are the bugchecks that cause OpenVMS to go down and reboot. The ERRFMT process reports bugchecks that do not cause the system to go down. The automated fix command to be executed when for these bugchecks are controlled by the ERRFMT_BUGCHECK_F parameter.

## CAPACITY_F=flags,command

This causes the specified command to be executed whenever the monitor process reports capacity-planning information.

## COMMAND#_F=flags,command

This causes the specified command to be executed anytime a user-defined command is executed which creates the output file to be reported by OSM. The # in the above parameter is any valid number in the range of 1 to 1000 and must match the COMMAND# parameter to which the fix is to be associated. For more information, refer to *Chapter 7: User-defined Event Monitoring* in this manual.

## COMPUTABLE_F=flags,command
## COMPUATABLE_CF=flags,command

This causes the specified command to be executed anytime a compute-bound process is detected and reported by the monitor process. The process ID of the process is passed as a parameter to this command.

### CPU_F=flags,command

This causes the specified command to be executed anytime the monitor processes reports that the CPU is spending an excessive amount of time in kernel mode, on the interrupt stack, or performing multiprocessing synchronization. Depending on what triggered the event, the string KERNEL, INTERRUPT or MPSYNC will be passed as a parameter to the subprocess executing the command to identify which mode of CPU activity triggered the event. If you have written a command procedure to fix this problem, you can perform different actions based on the nature of the problem.

### DEADLOCK_F=flags,command
### DEADLOCK_CF=flags,command

This causes the specified command to be executed anytime the monitor process reports an excessive rate of deadlock searches or detections.

### DISK_F=flags,command
### DISK_CF=flags,command

This causes the specified command to be executed whenever the monitor process reports disk usage exceeding the threshold. The name of the disk will be passed as a parameter to the subprocess executing the command so that the command (or command procedure) can know which disk triggered the event. These parameters may be set up on a per disk basis with the use of the OSM_DISK_xxx logical name. Refer to *Chapter 5: Controlling OSM Processes* in this manual for more information on OSM logical names.

### GBLSECTIONS_F=flags,command
### GBLSECTIONS_CF=flags,command

This causes the specified command to be executed whenever the monitor process reports that the number of free global pages or global sections are too low.

### IO_F=flags,command
### IO_CF=flags,command

This parameter causes the specified command to be executed whenever the monitor process reports excessive split I/Os or window turns.

### LOGICAL_NAMES_F=flags,command

This causes the specified command to be executed whenever the monitor process reports logical name performance statistics.

### NETWORK_ASCII_F=flags,command

This causes the specified command to be executed whenever the monitor process receives an ASCII message on the network.

### NETWORK#_F=flags,command

This causes the specified command to be executed anytime a remote node of the network is reported as being inaccessible. The # in the parameter is any valid number in the range of 1 to 1000 and must match the NETWORK# parameter to which the fix is to be associated.

### NETWORK_TIME#_F=flags,command

This causes the specified command to be executed anytime an event is reported due to differences in between the local node and a remote node. The # in the parameter is any valid number in the range of 1 to 1000 and must match the NETWORK_TIME# parameter to which the fix is to be associated.

### PAGEFILE_F=flags,command
### PAGEFILE_CF=flags,command

This causes the specified command to be executed whenever the monitor process reports that free page file or swap file space is too low. Either the string PAGE or SWAP will be passed as a parameter to subprocess executing the command (or command procedure) to let the command know if page file space or swap file space is too low.

### PAGE_FAULTS_F=flags,command

This causes the specified command to be executed whenever the monitor process reports excessive page faulting.

### POOL_F=flags,command
### POOL_CF=flags,command

This causes the specified command to be executed anytime excessive memory pool usage or pool allocation failures are reported by the monitor process.

### POWER_F=flags,command

This causes the specified command to be executed whenever the monitor process reports a power failure recovery.

### PROCESSES_F=flags,command
### PROCESSES_CF=flags,command

This causes the specified command to be executed anytime OSM reports a shortage of available process slots or balance set slots in OpenVMS.

### TEMPERATURE_F=flags,command
### TEMPERATURE_CF=flags,command

This causes the specified command to be executed anytime high temperatures are reported by the monitor process. The name of the temperature sensor is passed as a parameter to the command.

## XQP_CACHE_F=flags,command

This causes the specified command to be executed whenever the monitor process reports file system cache statistics.

## Error Format Automated Fix Parameters

This section lists parameters that cause the associated command to be executed upon the receipt of the associated error message from the ERRFMT process. Each of these parameters specifies the flags to be passed to the LIB$SPAWN routine and the command to be executed. The following table describes the error format automated fix parameters.

| Error Format Automated Fix Parameter Names | Description |
|---|---|
| ERRFMT_BUGCHECK_F | Nonfatal Bugcheck Errors |
| ERRFMT_BUS_F | External Bus Errors |
| ERRFMT_CARD_F | Card Reader Errors |
| ERRFMT_DISK_F | Disk Errors |
| ERRFMT_HARD_MEMORY_F | Hard Memory Errors |
| ERRFMT_INFORMATION_F | Informational Errors |
| ERRFMT_LINE_PRINTER_F | Line Printer Errors |
| ERRFMT_MAILBOX_F | Mailbox Errors |
| ERRFMT_MEMORY_F | Memory Errors |
| ERRFMT_MISCELLANEOUS_F | Miscellaneous Device Errors |
| ERRFMT_POWER_F | Power fail Recoveries |
| ERRFMT_REALTIME_F | Real-time Device Errors |
| ERRFMT_SYNCHRONOUS_ COMMUNICATION_F | Synchronous Communication |
| ERRFMT_SYSTEM_F | CPU, Console and Bus Errors |
| ERRFMT_TAPE_F | Tape Errors |
| ERRFMT_TERMINAL_F | Terminal Errors |
| ERRFMT_VOLUME_F | Volume Mounts and Dismounts |
| ERRFMT_WORKSTATION_F | Workstation Errors |

## Audit Server Automated Fix Parameters

These parameters cause the specified command string to be executed based on the receipt of the associated security alarm message from the AUDIT_SERVER process. Each of these parameters specifies the flags to be passed to the LIB$SPAWN routine and the command to be executed. The following table describes the audit server automated fix parameters.

| Audit Server Paging Parameter Names | Description |
|---|---|
| AUDIT_BREAKIN_F | Break-in Attempts |
| AUDIT_BYPASS_F | Use of Bypass Privilege to Access Files |
| AUDIT_INSTALL_F | Use of the Install Utility to Add or Remove a Known Image |
| AUDIT_LOGFAIL_F | Login Failures |
| AUDIT_READALL_F | Use of READALL Privilege to Access Files |
| AUDIT_UAF_F | SYSUAF Modifications Other Than a User Changing His Own Password Via Loginout or Set Password. |

# Chapter 12
# SYSUAF Synchronization

## Overview

SYSUAF (System Authorization File) maintenance can prove time consuming, especially when maintaining multiple SYSUAFs across many nodes. In essence, when dealing with a very large OpenVMS environment, where multiple changes are being made to SYSUAF records on many different systems, a large amount of time could be spent completing this task on all systems involved.

OSM can assist in maintaining the synchronization of the SYSUAFs on multiple nodes of a network. Synchronization is done by telling OSM on each node to notify a central distribution node of all changes to that node's SYSUAF. The central distribution node will then distribute the changes to other nodes, according to a distribution list. Those nodes will then apply the changes to their SYSUAFs.

In order to use the OSM interface, the OSM monitor process must be running on all nodes involved in the synchronization of SYSUAFs. Also, network communications must be set up for OSM. Please see the section in *Chapter 8: Network* Event Monitoring, for additional information about setting up network communication for OSM. OSM provides support for both DECnet and TCP/IP network communication. When attempting to communicate SYSUAF synchronization with another node, OSM will attempt first using DECnet, and if that fails, will try using TCP/IP.

# Features

- OSM simplifies the SYSUAF maintenance process, because there is no need to log into all nodes to update the SYSUAF, since they can be distributed from a centralized source.
- OSM can maintain a copy of all the SYSUAFs, for safety purposes, on the central distribution node. The SYSUAFs will be stored in a designated directory, created and maintained on the system. See an example in the parameters section of this chapter, which discusses the SYSUAF parameter.
- OSM will make changes to UIC-based identifiers that correspond to changes in the SYSUAF, by the use of SYSUAF_ALLOW discussed in the parameter section in this chapter.
- OSM uses the network to propagate changes to the SYSUAFs. The changes are distributed from a central distribution node to those nodes that are set up for synchronization.
- OSM provides a means to control which changes are propagated to each node via a distribution list. The distribution list is implemented through the use of logical names within OpenVMS. This distribution list specifies which fields of each SYSUAF record are propagated based on the username. It also allows preventing certain usernames from being propagated other nodes.

# Restrictions

- Direct writes to the SYSUAF that are made through applications other than OpenVMS supplied or supported utilities may not be propagated. All changes made through the use of the AUTHORIZE utility or the $SETUAI system service are supported.
- This feature is implemented by monitoring security messages from the AUDIT_SERVER process.
- OpenVMS V6.0 and V6.1 may not always create the required security message when a password is changed during login. In these cases, OSM will be unable to propagate the changed password to other systems. A patch has been created for this issue by OpenMS engineering. Once that patch has been installed, OSM works as intended.
- Using the SYSUAF_ALLOW parameter to restrict some types of modifications to a SYSUAF while having OSM maintain a copy of the SYSUAF on the central distribution node may cause the copy to become out of synch.

# Component Interaction

The following describes how the distribution list, SYSUAF synchronization, and the AUTHORIZE utility interact with each other:

- When a record is added to the SYSUAF, all fields in the record are propagated according to the distribution list. Any field that is not propagated is defaulted using the DEFAULT account record for that node. The UIC for that username will be added to the RIGHTSLIST database.
- When a record is copied in the SYSUAF, the fields are dealt with in the same way as an added record. The difference is any field that is not propagated is defaulted using the source record used in the copy. If the source record does not exist on the affected node, the record will not be copied. The UIC for that username will be added to the RIGHTSLIST database.
- When a record is renamed in the SYSUAF, the record will be renamed on all the nodes on which changes to any field would have been propagated. If any fields were changed with the rename, those changes will be propagated according to the distribution list. If the source record does not exist on the affected node, no changes are made. If there is an identifier for that username, that identifier will be renamed in the RIGHTSLIST database.
- When a record is modified in the SYSUAF, the record will be modified on all nodes based upon the distribution list. If the SYSUAF record does not exist on the target node, then no changes will be made. If the UIC is changed, and there is an identifier associated with that username, then the value of the identifier will be changed.
- When a record is removed from the SYSUAF, the record will be removed from all nodes on which changes to any field would have been propagated. If there is an identifier associated with the username that was removed, the identifier will also be removed.

# Distribution List

OSM utilizes a distribution list to control which fields are propagated for each username. This list is a set of logical name search lists. Logical names are used to allow the search list to be easily maintained and updated without requiring the OSM monitor process to be restarted. Much like access control lists in OpenVMS, OSM will use the first logical name translation that applies to a particular username.

For each set of usernames to be considered in the distribution of SYSUAF changes, there are two logical names that must be created. The first logical name, OSM_SYSUAF_USER_class_n, specifies the usernames to be affected. The second logical name, OSM_SYSUAF_FIELD_class_n, specifies which fields are to be propagated for the specified usernames. In both logical names, class is a name defined to represent a group of OpenVMS computers. The n in each logical name refers to a number relating a set of fields to a set of usernames. This number must begin with one and continue in order for all usernames to be defined. These logical names must be defined in the logical name table set by the LNM_TABLE parameter; or in the system logical name table if that parameter is not set.

To assist the system manager in setting up the distribution list, a template command procedure is provided with OSM. This procedure, OSM_SYSUAF_LOGICALS.TEMPLATE, may be copied to have a .COM extension and edited to meet the needs of the environment. This procedure defines all logical names in executive mode and in the table specified by the

LNM_TABLE parameter in OSM. If that parameter is not set, then the logical names are defined in the system logical name table. This allows the distribution list to be temporarily modified by defining a logical name with the in the same table, but without the /EXECUTIVE_MODE qualifier. The changes can then be restored by deassigning the supervisor mode logical name.

This procedure uses a DCL subroutine, DEFINE_LIST, to define the logical names described above. This subroutine has four parameters. The first parameter is the class name. If more than one class is defined as having the same distribution list, this may be specified as a list of class names separated by commas and enclosed in quotes. The second parameter is the sequence number for the logical name. These two parameters are used to make the name of the logical name. The third parameter is a list of usernames. The fourth parameter is a list of fields to be propagated for those usernames. Both parameters may be a comma-separated list enclosed in quotes. Any username or field name may include standard OpenVMS wildcard characters. For example, specifying “USER*, DEMO” for the username refers to all usernames starting with the characters USER or the username DEMO. If this procedure exists with a .COM extension, it will be executed upon the startup of OSM.

The distribution list can be complex, so examples are provided to assist the user in understanding how to properly set it up. In the examples, the actual DEFINE commands for the distribution list are omitted. Only the calls to the DEFINE_LIST subroutine are displayed.

**Note:** In the following examples there will be a central distribution node and all other nodes. All other nodes refer to the nodes other than the central distribution node that are involved with the SYSUAF synchronization.

## Example 1

There are two nodes, ALPHA and BETA, where ALPHA is the central distribution node. Both nodes are expected to have identical SYSUAF files. OSM will maintain this by propagating changes to all fields of all usernames to both nodes. On the central distribution node, the following logical names need to be defined in OSM_SYSUAF_LOGICALS.COM.

```
$ define osm_sysuaf_node alpha
$ define osm_sysuaf_class zone1
$ define osm_sysuaf_class_alpha alpha, zone1
$ define osm_sysuaf_class_beta beta, zone1
$ call define_list zone1 1 "*" "*"
```

Also, on the central distribution node in the OSM control file are some parameters that need to be set. The following list of parameters sets up ALPHA as the central distribution node, the order to distribute the SYSUAF changes, the wait time of 300 seconds to resend information to any of the nodes that may be down, and the ability for all SYSUAF transactions to be accepted. These parameters may be in either the node-specific parameter file, or the cluster-common parameter file.

```
sysuaf_notify=alpha
sysuaf_node=alpha
sysuaf_distribute_verify=300,alpha,beta
sysuaf_allow=all
```

All other nodes need the following logical name. This logical name is normally defined in OSM_LOGICALS.COM.

```
$ define osm_sysuaf_node beta
```

Also on all other nodes, the following parameters need to be set to allow propagation of all SYSUAF transactions.

```
sysuaf_notify=alpha
sysuaf_node=beta
sysuaf_allow=all
```

## Example 2

This example is the same as the above example, except it is set up to prohibit certain changes to SYSUAF records with usernames that begin with “TEST*”, but will allow changes to all other SYSUAF records to be propagated.

On the central distribution node, the same logical names need to be defined as in the above example, but another call line needs to be added for this example. This new call statement sets up which SYSUAF records are to be processed and which fields can be changed. In this case, any SYSUAF record for users “TEST*” will only be able to modify the password, wsdefault, wsquota, wsextent, astlm, biolm, diolm, enqlm, fillm, pbytlm, prclm, shrfillm and tqelm fields.

```
$ call define_list zone1 1 "test*" password,ws*,*lm
$ call define_list zone1 2 "*" "*"
```

All other settings would remain the same.

## Example 3

This example is the same as the one above, except it is set up to prohibit OSM from removing SYSUAF records. Everything remains the same as above, except the following parameter needs to be added to all nodes.

```
sysuaf_allow=all,noremove
```

All other settings remain the same.

## Example 4

There are three nodes, ALPHA, BETA and GAMMA in a cluster, which has a cluster alias of GREEK; as well as two standalone nodes, DELTA and ZETA. The cluster will be set up as the central distribution node. This cluster will maintain a copy of the SYSUAF from each node. This is set up to allow all transactions and changes to be made on all nodes. On the central distribution node, the following logical names need to be defined for this example:

```
$ define osm_sysuaf_node greek
$ define osm_sysuaf_class zone1
$ define osm_sysuaf_class_greek greek,zone1
$ define osm_sysuaf_class_alpha alpha,zone1
$ define osm_sysuaf_class_beta beta,zone1
$ define osm_sysuaf_class_gamma gamma,zone1
$ define osm_sysuaf_class_delta delta,zone2
$ define osm_sysuaf_class_zeta zeta,zone2
$ call define_list "zone1","zone2" 1 "*" "*"
```

The following list of parameters sets up the cluster as the central distribution node, the directory where the copies of SYSUAF are stored (%s will be substituted by OSM with the node names), and the nodes to distribute the SYSUAF changes. These parameters are put in the cluster common parameter file to allow OSM to handle the synchronization of SYSUAF changes when any one node of the cluster is accessible. This allows all SYSUAF transactions to be accepted.

```
sysuaf_notify=greek
sysuaf_node=greek
sysuaf=uaf_copy:sysuaf_%s.dat
sysuaf_distribute_verify=300,greek,delta,zeta
sysuaf_allow=all
```

All other nodes need to define the following logical name to identify them.

```
$ define osm_sysuaf_node nodename
```

The following parameters need to be set on all other nodes to allow propagation of all SYSUAF transactions. The nodename specified for the SYSUAF_NODE parameter is the same as the DECnet name for that node.

```
sysuaf_notify=greek
sysuaf_node=nodename
sysuaf_allow=all
```

## Example 5

In this example, we will be dealing with six nodes. They are ALPHA, BETA, GAMMA, DELTA, SIGMA and ZETA. Nodes ALPHA and BETA are in a cluster and share a common SYSUAF file. This cluster is the main production system and will be used as the central distribution node for SYSUAF synchronization. The cluster has an alias set up as GREEK. The remaining nodes are separate nodes in the network. Nodes GAMMA and DELTA are the development systems and are effectively identical to the production cluster. Nodes SIGMA and ZETA are Alpha development systems. The two VAX development systems should have identical SYSUAF files. The two Alpha development systems should have SYSUAF files identical to the VAX development systems except for the working set parameters. The development team has accounts on all systems that should be kept synchronized.

The central distribution nodes need to be set up with the following commands in OSM_SYSUAF_LOGICALS.COM.

```
$ define osm_sysuaf_class greek, software, alpha
$ define osm_sysuaf_class_greek greek, all, software, alpha
$ define osm_sysuaf_class_gamma develop, all, software, alpha
$ define osm_sysuaf_class_delta develop, all, software, alpha
$ define osm_sysuaf_class_sigma alpha, all, software
$ define osm_sysuaf_class_zeta alpha, all, software
$!
$! Class ALL will pass all changes to all fields to all nodes
$! This class is normally not used but is defined for those times
$! when it may be needed.
$ call define_list all 1 "*" "*"
$!
$! Class SOFTWARE is defined to pass all changes for the system account,
$! many changes to the accounts for the development team, and passwords
$! for all other usernames.
$ call define_list software 1 "jadams,msmith,pjones" -
  "acc*,device,directory,flags,owner,password,pwd*,uic,ws*"
$ call define_list software 2 system "*"
$ call define_list software 3 "*" "password,pwd*"
$!
$! Class ALPHA is the same as class SOFTWARE for the software
$! development team, except that working set fields are not propagated.
$! For all other accounts, only the password will be propagated.
$! fields are not updated for any accounts.
$ call define_list alpha 1 "jadams,msmith,pjones" -
  "acc*,device,directory,flags,owner,password,pwd*,uic"
$ call define_list alpha 2 "*" "password,pwd*"
```

The following parameters will need to be set up on nodes ALPHA and BETA. These parameters tell OSM where to distribute changes to SYSUAF files.

```
sysuaf_notify_verify=300,greek,gamma,delta,sigma,zeta
sysuaf_node=greek
sysuaf_allow=all
```

Nodes GAMMA and DELTA need the following logical name defined. This is normally defined in OSM_LOGICALS.COM.

```
$ define osm_sysuaf_class develop, software, alpha
```

Nodes SIGMA and ZETA need the following logical name defined.

```
$ define osm_sysuaf_class alpha, software
```

Also on all other nodes, the following parameters need to be set to allow propagation of all SYSUAF modifications.

```
sysuaf_notify=alpha,beta
sysuaf_node=gamma
sysuaf_allow=all
```

# Logical names

In order to use SYSUAF synchronization, the OSM monitor process must be running on all nodes involved. In addition, network communication must be set up for OSM. See Network Communications section in *Chapter 5: Controlling OSM Processes* for additional information about setting up network communications.

This section describes the logical names used by OSM in setting up synchronization of SYSUAFs. For all nodes other than the central distribution node, OSM_SYSUAF_CLASS is the only logical name that must be defined. All logical names are to be defined on the central distribution node.

## OSM_SYSUAF_CLASS

By default, this logical name is defined as the translation of the logical name SYS$NODE during the startup of OSM. This logical may be defined as either a cluster alias, the node name, or a string designating a class to which this system belongs. This may also be defined as a logical name search list to include one node in up to 8 different classes. If this is done, the first translation of this logical name is considered the primary class for this node.

If a node should be ignored by SYSUAF synchronization for a particular change, then on that node, define this logical name to a nonexistent class, make the desired change in AUTHORIZE, and restore the logical name. This will tell OSM that this node will not be using the SYSUAF synchronization while this logical name is defined.

## OSM_SYSUAF_CLASS_nodename

This logical name will be defined on the central distribution node to cover each desired DECnet node on the network. Every node participating in the SYSUAF synchronization must be defined, as shown below. The first translation of this logical name for each node specifies the primary class to which the node belongs. Any changes made on a node with its OSM_SYSUAF_CLASS logical name defined to the primary class will be propagated without restriction to the specified node. All other translations of the logical name are secondary classes to which the node belongs. Changes to the SYSUAF on nodes in these classes are propagated according to the distribution list.

```
$ define osm_sysuaf_class_alpha alpha,abc
$ define osm_sysuaf_class_beta beta,abc
$ define osm_sysuaf_class_gamma gamma,abc
```

The examples above show that the abc class designates nodes ALPHA, BETA and GAMMA.

## OSM_SYSUAF_FIELD_class_n

This logical name defines a set of fields to a particular class. It is used in conjunction with the OSM_SYSUAF_USER_class_n logical names to define the distribution list.

The DCL subroutine, DEFINE_LIST, will create both the OSM_SYSUAF_FIELD_class_n and the OSM_SYSUAF_FIELD_class_n logical names in OSM_SYSUAF_LOGICALS.TEMPLATE. The field name(s) may include standard OpenVMS wildcard characters. * will match all field names. The following table lists the valid field names.

| Name | Description |
|---|---|
| ACCESS | Hours of access for all modes of access |
| ACCOUNT | Name for the account |
| ASTLM | AST queue limit |
| BATCH | Time of access permitted for batch jobs |
| BIOLM | Buffered I/O count limit |
| BYTLM | Buffered I/O byte limit |
| CLI | Name of the default command language interpreter |
| CLITABLES | Name of the CLI tables |
| CPUTIME | Maximum process CPU time |
| DEFPRIVILEGES | Privileges enabled upon login |
| DEVICE | Default device |
| DIALUP | Dialup access restrictions |
| DIOLM | Number of direct I/Os that can be outstanding |
| DIRECTORY | Default directory |
| ENQLM | Number of locks that can be queued at one time |
| EXPIRATION | Expiration date for the account |
| FILLM | Maximum number of files that can be opened at one time |
| FLAGS | Login flags for the user |
| INTERACTIVE | Time of access for interactive logins |
| JTQUOTA | Quota for the job-wide logical name table |
| LGICMD | Default login command file |
| LOCAL | Time of access for interactive logins from local terminals |
| MAXDETACH | Maximum number of detached jobs active at one time |
| MAXJOBS | Maximum number of jobs active at one time per username |
| MAXACCTJOBS | Maximum number of jobs active at one time per account |
| NETWORK | Hours of network access for network jobs |
| OWNER | Name of the owner of the account |

| Name | Description |
|---|---|
| PASSWORD | Password |
| PBYTLM | Maximum number of bytes of paged pool<br>(Not implemented in all versions of OpenVMS) |
| PFLAGS | Primary Restrict Flags |
| P_RESTRICT | Primary Access Restrict |
| PGFLQUOTA | Paging file limit |
| PRCLM | Subprocess creation limit |
| PRIMEDAYS | Primary and secondary days for logging into system |
| PRIORITY | Default base priority |
| PRIVILEGES | Privileges users are authorized to enable |
| PWDLIFETIME | Length of time a password will remain valid |
| PWDMINIMUM | Required number of characters in a password |
| QUEPRIORITY | Maximum priority a for queuing operation<br>(Not implemented in all versions of OpenVMS) |
| REMOTE | Time of access permitted for interactive logins from network remote terminals |
| SFLAGS | Secondary Restrict Flags |
| S_RESTRICT | Secondary Access Restrict |
| SHRFILLM | Maximum number of shared files a user can have open at one time |
| TQELM | Total number of timer queue entries plus number of temporary common event flag clusters that a user can have at one time |
| UIC | User Identification Code |
| WSDEFAULT | Default working set size |
| WSEXTENT | Working set extent |
| WSQUOTA | Working set quota |

## OSM_SYSUAF_USER_class_n

This logical name allows defining a set of users to a particular class. It is used in conjunction with the OSM_SYSUAF_FIELD_class_n logical name. This is defined as one or more usernames. The usernames may include standard OpenVMS wildcard characters to select multiple usernames.

# SYSUAF Synchronization Commands

The following are the OSM commands that are used to view and clear SYSUAF synchronization information.

## CLEAR SYSUAF

This command clears pending SYSUAF notification or distribution information from the OSM synchronization data. To determine what SYSUAF synchronization information is pending, issue the OSM command SHOW STATUS/SYSUAF.

**Format**

**CLEAR SYSUAF**

**Qualifiers**

**/ALL (default)**

Specifies that the requested information is to be cleared from both the distribute and notify data. This is the same as specifying both /NOTIFY and /DISTRIBUTE.

**/DISTRIBUTE**

Specifies that this command is to clear SYSUAF distribution messages. The default is to clear both notification and distribution messages.

**/NODE=***

Specifies the name of the nodes to which the messages are to be sent. The default is to clear messages to all nodes.

**/NOTIFY**

Specifies that this command is to clear SYSUAF notification messages. The default is to clear both notification and distribution messages.

**/USERNAME=***

Specifies the usernames related to the SYSUAF synchronization data to be sent to another node. The default is to remove messages related to all usernames.

## SHOW STATUS

This command allows a user to see the current status of OSM. When combined with the /SYSUAF qualifier, this includes SYSUAF synchronization information. For additional information on this command, and an example of its output, refer to *Chapter 5: Controlling OSM Processes*.

**Qualifiers**

**/FULL**

When combined with /SYSUAF, /FULL displays the name of each user record to be synchronized. If multiple changes have been specified, then the same user record may be displayed multiple times.

**/SYSUAF**

Includes SYSUAF synchronization information in the output.

## SYNCHRONIZE

This command notifies OSM to synchronize values of an account in the SYSUAF with other nodes running OSM. Synchronization is done as though the changes were just made to the account and is based on the configuration.

Qualifiers specify which fields in the user record are flagged as needing synchronization. By default, all fields in the record are synchronized.

**Format**

SYNCHRONIZE username /qualifier[,...]

**SYNCHRONIZE username**

**Parameter**

**Username**

Specifies the name of a user in the SYSUAF. The asterisk (*) and percent sign (%) wildcard characters are permitted in the username. When you specify a single asterisk for the username, you modify the records of all users.

**Qualifiers**

**/ALL**

Specifies that all attributes of the account(s) should be synchronized based on the rules configured for OSM. This is the default if no other qualifiers are specified.

**/ACCESS**

Specifies hours of access for all modes of access.

**/ACCOUNT**

Specifies the name for the account. This can be a string of 1 to 8 alphanumeric characters.

**/ASTLM**

Specifies the AST queue limit, which is the total number of asynchronous system trap (AST) operations and scheduled wake-up requests that the user can have queued at one time.

**/BATCH**

Specifies the hours of access permitted for batch jobs.

**/BIOLM**

Specifies a buffered I/O count limit for the BIOLM field of the UAF record. The buffered I/O count limit is the maximum number of buffered I/O operations, such as terminal I/O, that can be outstanding at one time.

**/BYTLM**

Specifies the buffered I/O byte limit for the BYTLM field of the UAF record. The buffered I/O byte limit is the maximum number of bytes of nonpaged system dynamic memory that a user's job can consume at one time. Nonpaged dynamic memory is used for operations such as I/O buffering, mailboxes, and file-access windows.

**/CLI**

Specifies the name of the default command language interpreter (CLI) for the UAF record. This a string of 1 to 31 alphanumeric characters. This setting is ignored for network jobs.

**/CLITABLES**

Specifies user-defined CLI tables for the account. The filespec can contain 1 to 31 characters. This setting is ignored for network jobs.

**/CPUTIME**

Specifies the maximum process CPU time for the CPU field of the UAF record. The maximum process CPU time is the maximum amount of CPU time a user's process can take per session. The default value of 0 means an infinite amount of time.

**/DEFPRIVILEGES**

Specifies default privileges for the user; that is, those enabled at login time.

**/DEVICE**

Specifies the name of the user's default device at login. The device-name is a string of 1 to 31 alphanumeric characters. Logical names may be used.

**/DIALUP**

Specifies hours of access permitted for dialup logins.

**/DIOLM**

Specifies the direct I/O count limit for the DIOLM field of the UAF record. The direct I/O count limit is the maximum number of direct I/O operations (usually disk) that can be outstanding at one time.

**/DIRECTORY**

Specifies the default directory name for the DIRECTORY field of the UAF record. The directory-name can be 1 to 39 alphanumeric characters.

**/ENQLM**

Specifies the lock queue limit for the ENQLM field of the UAF record. The lock queue limit is the maximum number of locks that can be queued by the user at one time.

**/EXPIRATION**

Specifies the expiration date and time of the account.

**/FILLM**

Specifies the open file limit for the FILLM field of the UAF record. The open file limit is the maximum number of files that can be open at one time, including active network logical links.

**/FLAGS**

Specifies login flags for the user.

**/INTERACTIVE**

Specifies the hours of access for interactive logins.

**/JTQUOTA**

Specifies the initial byte quota with which the job-wide logical name table is to be created.

**/LGICMD**

Specifies the name of the default login command file. The file name defaults to the device specified for /DEVICE, the directory specified for /DIRECTORY, a file name of LOGIN, and a file type of .COM.

**/LOCAL**

Specifies hours of access for interactive logins from local terminals.

**/MAXACCTJOBS**

Specifies the maximum number of batch, interactive, and detached processes that can be active at one time for all users of the same account. A value of 0 represents an unlimited number.

**/MAXDETACH**

Specifies the maximum number of detached processes with the cited username that can be active at one time. A value of 0 represents an unlimited number.

**/MAXJOBS**

Specifies the maximum number of processes (interactive, batch, detached, and network) with the cited username that can be active simultaneously. The first four network jobs are not counted. A value of 0 represents an unlimited number.

**/NETWORK**

Specifies hours of access for network batch jobs.

**/OWNER**

Specifies the name of the owner of the account. The owner name is 1 to 31 characters.

**/PASSWORD**

Specifies both passwords for login and associated fields. Passwords can be from 0 to 32 alphanumeric characters in length. The dollar sign ($) and underscore (_) are also permitted.

**/PBYTLM**

This flag is reserved for HP.

**/PGFLQUOTA**

Specifies the paging file limit. This is the maximum number of pages that the person's process can use in the system paging file.

**/PRCLM**

Specifies the subprocess creation limit. This is the maximum number of subprocesses that can exist at one time for the specified user's process.

**/PRIMEDAYS**

Defines the primary and secondary days of the week for logging in. Use the primary and secondary day definitions in conjunction with such qualifiers as /ACCESS, /INTERACTIVE, and /BATCH.

**/PRIORITY**

Specifies the default base priority. The value is an integer in the range of 0 to 63 on Alpha and Integrity server systems or 0 to 31 on VAX systems.

**/PRIVILEGES**

Specifies which privileges the user is authorized to hold, although these privileges are not necessarily enabled at login. The /DEFPRIVILEGES qualifier determines which ones are enabled.

**/PWDLIFETIME**

Specifies the length of time a password is valid.

**/PWDMINIMUM**

Specifies the minimum password length in characters. Note that this value is enforced only by the DCL command SET PASSWORD.

**/QUEPRIO**

Reserved for future use.

**/REMOTE**

Specifies hours during which access is permitted for interactive logins from network remote terminals (with the DCL command SET HOST).

**/SHRFILLM**

Specifies the maximum number of shared files that the user can have open at one time.

**/TQELM**

Specifies the total number of entries in the timer queue plus the number of temporary common event flag clusters that the user can have at one time.

**/UIC**

Specifies the user identification code (UIC). The UIC value is a group number in the range from 1 to 37776 (octal) and a member number in the range from 0 to 177776 (octal), which are separated by a comma and enclosed in brackets.

**/WSDEFAULT**

Specifies the default working set limit. This represents the initial limit to the number of physical pages the process can use.

**/WSEXTENT**

Specifies the working set maximum. This represents the maximum amount of physical memory allowed to the process. The system provides memory to a process beyond its working set quota only when it has excess free pages. The additional memory is recalled by the system if needed.

**/WSQUOTA**

Specifies the working set quota. This is the maximum amount of physical memory a user process can lock into its working set. It also represents the maximum amount of swap space that the system reserves for this process and the maximum amount of physical memory that the system allows the process to consume if the system-wide memory demand is significant.

**Examples**

**$ OSM SYNCHRONIZE WILLIAMS**

This command synchronizes all fields for the WILLIAMS username with other nodes configured for OSM SYSUAF synchronization.

**$ OSM SYNCHRONIZE *_TEST /BYTLM /BIOLM**

This command synchronizes the user's byte limit and buffered I/O limit for all usernames ending in _TEST with other nodes configured for OSM SYSUAF synchronization.

# SYSUAF Synchronization Parameters

## SYSUAF=file-specification

This parameter specifies the device, directory, and file name for the copies of the SYSUAFs on the central distribution node. The default is a null string, which does not maintain copies of the SYSUAF on the central distribution node. The following format is used to set up the SYSUAF files in one directory on the central distribution node:

```
sysuaf=uaf_copy:sysuaf_%s.dat
```

The %s will be replaced by OSM with the names of the nodes to receive distributions. If this parameter is defined, and the specified files exist, then OSM will update the files as it distributes changes to the SYSUAF.

## SYSUAF_ALLOW=options

This controls which options may be propagated from a remote node to this node. This should be set up on all nodes. By default, it is set to null which will allow no transactions. The valid options for this parameter are listed in the following table.

| Keyword | Action |
|---|---|
| ALL | Specifying this keyword enables all other options listed below. It is possible to specify this keyword followed by the negated version of one of the following options. Example: SYSUAF_ALLOW=ALL,NOREMOVE |
| ADD | This allows records added through the authorize utility on a remote node to be propagated to this node. |
| COPY | This allows records copied through the authorize utility on a remote node to be propagated to this node. |
| MODIFY | This allows records modified through the authorize utility on a remote node to be propagated to this node. |
| REMOVE | This allows records removed through the authorize utility on a remote node to be propagated to this node. |
| RENAME | This allows records renamed through the authorize utility on a remote node to be propagated to this node. |
| UIC_IDENTIFIER | This allows OSM to update UIC-based identifiers that correspond to changes in SYSUAF. |

### SYSUAF_DISTRIBUTE=nodename,...
### SYSUAF_DISTRIBUTE_ADD=nodename,...
### SYSUAF_DISTRIBUTE_VERIFY=0,nodename,...
### SYSUAF_DISTRIBUTE_VERIFY_ADD=nodename,...

The SYSUAF_DISTRIBUTE parameter is a list of all the nodes that should be notified. An attempt will be made to notify each node once. The SYSUAF_DISTRIBUTE_ADD parameter adds node names to the list set up by the SYSUAF_DISTRIBUTE parameter. This allows the user to set up a large list of nodenames to which SYSUAF modifications are to be distributed.

The SYSUAF_DISTRIBUTE_VERIFY parameter is a list of all the nodes that should be notified. OSM will attempt to notify a node until it is successful. The number specified for this parameter is the seconds between retry attempts, for any node that OSM was unsuccessful in notifying. Notification of a node should be done by one parameter or the other, but not both. The SYSUAF_DISTRIBUTE_VERIFY_ADD parameter adds node names to the list set up by the SYSUAF_DISTRIBUTE_VERIFY parameter. The SYSUAF_DISTRIBUTE_VERIFY_ADD parameter does not include the number of seconds between retry attempts. This allows the user to set up a large list of nodenames to which SYSUAF modifications are to be distributed.

### SYSUAF_NODE=nodename

This parameter can be defined as a nodename or a cluster alias. By default, it will be the logical name translation of SYS$CLUSTER_NODE or if that is not defined it will be the translation of the logical name SYS$NODE.

### SYSUAF_NOREMOVE=SYSTEM,DEFAULT

This parameter prevents the removal a user or a list of users on a remote node from being propagated to this node. This default value of this parameter prevents OSM from removing or renaming either the SYSTEM or DEFAULT accounts. The usernames may include standard OpenVMS wildcard characters.

### SYSUAF_NOTIFY=nodename...

This parameter specifies the node name of the central distribution node, or a list of node names that will serve as the central distribution node, which should be notified via DECnet. The first node on the list will be notified if it is available when a notification request is made and then OSM will continue to notify all other nodes on the list if they are available. The two OSM parameters, DECNET_OBJECT=300,OSM_MON and AUDIT =Y must be set on this node. Both are enabled by default. This parameter must be set up on all nodes.

# Chapter 13
# OSM Testing

## Overview

This section describes the DEMONSTRATE commands and the syntax for each one. Several commands exist to allow you to test the creating and reporting of an event. These commands can be used to test and verify that the parameters for OSM that you set up in your environment were done correctly to achieve the results that you intended.

Once you have set up OSM to monitor the system, you may wish to execute some of these commands to verify that the reports OSM generates are the ones you want. If you set up OSM to page you when a specific event occurs, you can use the demonstrate commands to verify that you set everything up correctly to be paged. If you set up OSM to execute an automated fix command, you may use these commands to verify that the automated fixes will execute.

Each of the following DEMONSTRATE commands affect an area of the system which may be monitored by OSM. Since all areas of monitoring by OSM are controlled by parameters, it is possible to affect whether or not an event is reported by any one command. If an exception event is reported, the severity of that event will control whether it is reported immediately, or the reporting of the event is delayed until the following night. If it is not a critical event, it is possible to expedite the reporting of that event by executing the SEND ALL command within OSM.

# Demonstrate commands

## DEMONSTRATE ALIGNMENT_FAULTS

This command generates alignment faults for the specified period of time.

**Format**

**DEMONSTRATE ALIGNMENT_FAULTS**

**Qualifiers**

**/END=delta-time**

Specifies the time which this command will stop executing. The time specified may be either an absolute time or a combination time. To specify a delta time, enter it as a combination time but default the current time. The default delta time is two times the CPU_WAKE parameter. This command may be aborted early by pressing control/C.

**/EXECUTIVE_MODE**

Causes the command to generate the alignment faults in executive mode. The default is to generate user mode alignment faults.

**/FAULTS=count**

Specifies the number of alignment faults to generate each second. The default is to generate 1000 alignment faults per second.

**/KERNEL_MODE**

Causes the command to generate the alignment faults in kernel mode. The default is to generate user mode alignment faults.

**/USER_MODE**

Causes the command to generate the alignment faults in user mode. This is the default.

## DEMONSTRATE AUDIT_SERVER

This command copies existing security audit records from an audit server log file to OSM. The file specified may include security alarms gathered on any system. OpenVMS keeps security alarms in SYS$MANAGER:SECURITY_AUDIT.AUDIT$JOURNAL. This command is valid only for OpenVMS systems.

**Format**

**DEMONSTRATE AUDIT_SERVER file-name**

**Parameter**

**file-name**

You must specify the name of the audit server log file or a copy of that file that you may have created. There is no default for this parameter.

**Qualifiers**

**/DELAY=seconds**

Specifies the number of seconds that should pass between each record written to the OSM monitor process. The default is no delay.

**/ENTRY=(START:1,END:2147483647)**

Specifies the starting and ending records to be copied from the specified file to OSM. The default sends the whole file. The parentheses are required if both the starting and ending records are specified.

## DEMONSTRATE BREAKIN

This command creates a break-in attempt on the local node by attempting to access a file on the system disk using a nonexistent account. This command requires that DECnet be installed and running. For the OSM monitor process to detect either the login failure or break-in attempt, the OSM parameter AUDIT must not be set to N.

**Format**

**DEMONSTRATE BREAKIN**

**Qualifiers**

**/ATTEMPTS=retry-count**

Specifies the number of times an attempt is made to access the file. This command will default to attempt one more time than the system parameter LGI_BRK_LIM. Specifying /ATTEMPTS=1 will cause a single login failure to be reported by the operating system to OSM.

**/USERNAME=123ABCXQZ**

This specifies the username to be used for the break-in attempts. This should be an account that does not exist on the system.

## DEMONSTRATE BUGCHECK

This command creates a user-mode bugcheck that will be logged by the ERRFMT process. It requires that the process executing this command be authorized to have the BUGCHK privilege enabled. This bugcheck will not affect the operation of the system or the process. The bugcheck code created is PROCGONE, Process not in system.

For the OSM monitor process to detect this event, the parameter ERRFMT must not be set to N. The ERRFMT process may take up to 30 seconds to report this event to the OSM monitor process.

**Format**

**DEMONSTRATE BUGCHECK**

## DEMONSTRATE CPU

This command creates high kernel mode CPU activity for the specified time.

**Format**

**DEMONSTRATE CPU**

**Qualifiers**

**/END=delta-time**

Specifies the time which this command will stop executing. The time specified may be either an absolute time or a combination time. To specify a delta time, enter it as a combination time but default the current time. The default delta time is two times the CPU_WAKE parameter. This command may be aborted early by pressing control/C.

**/PRIORITY=priority**

Specifies as scheduling priority level from 0 to 31 for which the command should execute. If the priority is set too high on a heavily loaded CPU with insufficient resources, it may interfere with other users on the system. If the priority is set to 16 or higher, then only this command will execute on one processor. In a multiprocessor environment, other processes may continue executing on other processors. If the priority is set too low, it may not have time enough to create an event. The default is to execute at the user's current base priority.

## DEMONSTRATE DEAD_LOCKS

This command creates deadlocks that will be detected and reported by OSM.

**Format**

**DEMONSTRATE DEAD_LOCKS**

**Qualifiers**

**/ATTEMPTS=attempt-count**

This specifies the number of deadlocks per second that this command should create. The default for this parameter is created by the OSM parameter MONITOR_WAKE multiplied by the DEADLOCK parameter for the number of deadlocks found times two. If these parameters are at their default value, this command will attempt to generate 60 deadlocks per second.

## DEMONSTRATE DISK dev-name

This command attempts to create events on disk activity for a specified period of time. You must specify either the /IO_RATE or the /FREE qualifiers or the command will not have any affect.

**Format**

**DEMONSTRATE DISK dev-name**

**Parameter**

**dev-name**

Specifies the name of the disk on which events are to be created. The disk must be mounted. You can use either the device name of the disk or a logical name assigned to it.

**Qualifiers**

**/END=delta-time**

Specifies the time which this command will stop executing. The time specified may be either an absolute time or a combination time. To specify a delta time, enter it as a combination time but default the current time. The default delta time is two times the MONITOR_WAKE parameter. This command may be aborted early by pressing control/C.

**/FREE=max-free-blocks**

The /FREE qualifier creates a temporary file on the disk with a size that will leave the specified number of blocks free. It will be left there until the specified end time, after which it will be deleted. This gives the OSM monitor enough time to detect the lack of free space on the disk and report it. When using the /FREE qualifier you must specify a disk that is mounted and is write enabled.

**/IO_RATE=io-per-sec**

This qualifier causes OSM to attempt to create I/O activity on the specified disk at the desired rate. This is done by logical read I/Os to block 0 of the disk. Normally this is the home block and does not contain any user data. It also requires the process to have LOG_IO privilege.

**Note:** Setting the I/O rate over 100 I/Os per second may be too high depending on the disk. This could cause severe performance problems on some systems. This problem will only occur during the execution of this command and normal operation will resume once the command terminates.

## DEMONSTRATE ERRORLOG file-name

This command copies existing errorlog entries from the specified errorlog file to OSM.

**Format**

**DEMONSTRATE ERRORLOG file-name /qualifier**

**Parameter**

**file-name**

Specifies the name of the file from which the errorlog entries are to be copied. You may specify any errorlog file created on any system of the same architecture as the one on which the command is executed. OSM provides three errorlog files for this purpose. Please choose one that is appropriate for your system. ERRLOG_VAX.SYS, ERRLOG_ALPHA.SYS and ERRLOG_I64.SYS reside in the OSM$ directory.

**Qualifiers**

**/DELAY=seconds**

Specifies the number of seconds that should pass between each record written to the OSM monitor process. The default is no delay.

**/ENTRY=(START:1,END:2147483647)**

This is to specify the starting and ending record numbers. The default is to send the whole file. The parentheses are required only if both the starting and ending record numbers are specified.

## DEMONSTRATE GLOBAL_SECTIONS

This command reduces the number of available global sections by creating temporary global sections containing one page each. It leaves those sections for the specified duration of time.

**Format**

**DEMONSTRATE GLOBAL_SECTIONS**

**Qualifiers**

**/END=delta-time**

Specifies the time which this command will stop executing. The time specified may be either an absolute time or a combination time. To specify a delta time, enter it as a combination time but default the current time. The default delta time is two times the MONITOR_WAKE parameter. This command may be aborted early by pressing control/C.

**/SECTIONS=count**

Specifies the number of global sections to leave free. The default for this qualifier is 2% of the system parameter GBLSECTIONS.

## DEMONSTRATE IO

This command attempts to create a high split I/O rate and a high window turn rate.

**Format**

**DEMONSTRATE IO**

**Qualifiers**

**/END=delta-time**

Specifies the time which this command will stop executing. The time specified may be either an absolute time or a combination time. To specify a delta time, enter it as a combination time but default the current time. The default delta time is two times the MONITOR_WAKE parameter. This command may be aborted early by pressing control/C.

**/ATTEMPTS=count**

Specifies the minimum number of split I/O and window turns that should be attempted each second. The default is 1 greater than the larger of the thresholds set for window turns and split I/O as specified by the IO parameter.

## DEMONSTRATE LOGICAL_NAMES

This command attempts to create the specified number of shared logical names in one logical name hash chain. It leaves those logical names for the specified duration of time.

**Format**

**DEMONSTRATE LOGICAL_NAMES**

**Qualifiers**

**/END=delta-time**

Specifies the time which this command will stop executing. The time specified may be either an absolute time or a combination time. To specify a delta time, enter it as a combination time but default the current time. The default delta time is two times the MONITOR_WAKE parameter. This command may be aborted early by pressing control/C.

**/LOGICAL_NAMES=count**

Specifies the number of shared logical names to be in one logical name hash chain. The default for this qualifier is 70 logical names.

## DEMONSTRATE NETWORK_OBJECT

This command sends an ASCII message to the OSM monitor process on the specified node.

**Format**

**DEMONSTRATE NETWORK_OBJECT message-text**

**Parameter**

**message-text**

Specifies the text of the message to send to the remote node. The text sent in the message is the remainder of the line. All qualifiers to this command must be specified before the message-text.

**Qualifiers**

**/DECNET**

Specifies that this command should use DECnet as the communication path between the local node and the remote node. If neither DECNET nor TCPIP are specified, then the network used is controlled by the DEFAULT_NETWORK parameter. This requires that DECnet be installed and set up on both the local system and the system specified by the NODE qualifier.

**/NODE_NAME=node-name**

Specifies the name of the node to which the message should be sent. The default is to send it to the OSM monitor process on the local node.

**/SEVERITY=severity**

Specifies the severity of the message sent. The default is to send it at the severity of the NETWORK_ASCII parameter on the remote node.

**/TCPIP**

Specifies that this command should use TCP/IP as the communication path between the local node and the remote node. If neither DECNET nor TCPIP are specified, then the network used is controlled by the DEFAULT_NETWORK parameter. This requires that a supported TCP/IP software package be installed and set up on both the local system and the system specified by the NODE qualifier. See the *Software Requirements* section in *Chapter 1: Introduction to OSM* for a list of supported implementations of TCP/IP.

## DEMONSTRATE PAGE_FAULTS

This command attempts to create the specified page fault rate for the duration of time specified. The number of page faults per second generated should be as many as requested, but may be more, as long as the CPU processing power supports it for a single process.

**Format**

**DEMONSTRATE PAGE_FAULTS**

**Qualifiers**

**/END=delta-time**

Specifies the time which this command will stop executing. The time specified may be either an absolute time or a combination time. To specify a delta time, enter it as a combination time but default the current time. The default delta time is two times the CPU_WAKE parameter. This command may be aborted early by pressing control/C.

**/FAULTS=faults_per_second**

Specifies the desired number of page faults per second. This should be higher than the OSM parameter PAGE_FAULTS. The default number of page faults per second is 50.

**/SYSTEM**

This generates system faults instead of process faults. It does this by referencing paged pool. This parameter is incompatible with /FAULTS.

## DEMONSTRATE PAGR

This command generates a page and sends that page to the transport process to be invoked. Note that PAGR does not include the letter E to prevent conflicts with the DEMONSTRATE PAGE_FAULTS command. See *Chapter 10: OSM Paging Support* for additional information about pager notification strings. The user must specify both the message and the /MAIL qualifier.

**Format**

**DEMONSTRATE PAGR message**

**Parameter**

**message**

Specifies the text message to be sent to the pager.

**Qualifiers**

**/MAIL=pager-string**

Specifies the pager notification string for the event.

## DEMONSTRATE XQP_CACHE

This command attempts to create a low hit rate on the file header cache.

**Format**

**DEMONSTRATE XQP_CACHE**

**Qualifiers**

**/END=delta-time**

Specifies the time which this command will stop executing. The time specified may be either an absolute time or a combination time. To specify a delta time, enter it as a combination time but default the current time. The default delta time is two times the MONITOR_WAKE parameter. This command may be aborted early by pressing control/C.

**/ATTEMPTS=count**

Specifies the number of attempts that should be made to the file header cache each second. The default is 1 greater than the attempt rate specified in the XQP_CACHE parameter.

# Chapter 14
# Monthly Summary Reporting

## Overview

OSM provides the ability to generate a monthly summary of exception events, performance statistics, disk space utilization and error log information. Basically, the summary feature allows you to generate a report summarizing everything on which OSM has monitored and reported.

The OSM SUMMARIZE command generates this summary report. For the SUMMARIZE command to work, all binary event files for the period to be summarized must be saved.

To help you implement the summary feature; two command procedures have been provided. One causes OSM to save all the binary event files, and the other can be set up as a User-Defined Monitoring (UDM) command procedure. For more information on User-defined Event Monitoring, reference *Chapter 7: User-defined Event Monitoring* in this manual.

The two command procedures provided which assist in the implementation of summary reports for OSM data are:

### OSM_SUMMARY_SETUP.COM

This command procedure only needs to be executed once to set up retention of the binary event files to summarize. It verifies that the OSM parameter BINARY is included in the options for the OSM parameter OUTPUT. It will also verify that the BINARY parameter is specified and that it points to a valid directory. If this parameter is not specified, then it adds the parameter BINARY=OSM_SUMMARY: and creates an OSM_SUMMARY subdirectory to the OSM$ directory. It will then define the OSM_SUMMARY logical name to point to that directory. The next execution of OSM_STARTUP.COM will recreate that logical name for each reboot of the system.

### OSM_SUMMARY_UDM.TEMPLATE

To utilize this template file, it should be copied to OSM_SUMMARY_UDM.COM, edited for your environment and set up as a user-defined monitored event to be executed multiple times a day. When executed, this procedure determines if it is the first of the month. If so, it will submit itself to a batch queue to be executed the following night to generate the summary report. The template executes multiple times a day so that it has a better chance of executing on the first of the month if your system(s) might be down for any reason. This procedure assumes that summary information is to be created once each month for the previous month's data.

## Setting up Monthly Summary Reporting

The following are the steps necessary to set up monthly summary reports from OSM:

1) Execute the OSM_SUMMARY_SETUP.COM command procedure. This procedure causes OSM to save all binary event files needed to generate a summary report.
2) Copy the file OSM_SUMMARY_UDM.TEMPLATE to OSM_SUMMARY_UDM.COM and edit it to include any modifications desired. Modifications may include printing the report or sending the report via MAIL to the system manager. It could also include having it delete previously processed files that are no longer needed.
3) Setup User-defined Monitoring to detect the end of month. This is accomplished by placing the following line in the desired OSM parameter file (typically the node-specific parameter file):

```
COMMAND#=28800,0,@OSM_SUMMARY_UDM.COM CHECK SYS$BATCH
```

The following describes the components of the line placed in the parameter file. For more information, refer to *Chapter 7: User-defined Event Monitoring* in this manual, or the OSM_SUMMARY_UDM.TEMPLATE file.

#
This must be a unique number within the range of 1-1000 that is used to identify the User-defined event Monitoring (UDM) command.

28800
The number of seconds that must pass before the UDM re-executes. This is equivalent to 8 hours. You may change this value to the interval you desire.

0
These are the spawn flags. The value of 0 indicates that OSM perform a SPAWN/WAIT to execute the specified command procedure.

@OSM_SUMMARY_UDM.COM
This is the UDM command executed. In this case, it's the name of the command procedure that determines if it is the first of the month, and if so, generates the OSM summary report.

CHECK SYS$BATCH
These are parameters passed to the UDM command procedure. In this case, CHECK indicates that the command procedure should check to see if it is the first of the month. If it is the first, it checks if the summary command procedure is currently in the SYS$BATCH queue for processing. If it's the first of the month, but not in the queue, it will submit the summary command procedure to the queue. If it's not the first of the month, it exits.

4) Stop and restart the OSM monitor and transport processes. This is needed to cause both processes to load the new parameters.

# OSM Summary Commands

The OSM SUMMARIZE command is the only OSM command that is used in summarizing OSM data. It has the following two formats:

## SUMMARIZE/OSM

The SUMMARIZE command has two formats. SUMMARIZE/OSM processes saved binary event files to generate a report summarizing all data collected by OSM. This is the default format for the SUMMARIZE command and therefore does not require the /OSM qualifier.

**Format**

**SUMMARIZE/OSM input-files**

**Parameters**

**input-files**

Specifies the name of the input file(s) to be summarized. This is usually OSM_BINARY:OSM_DATAFILE_*.SEND;*. There is no default, so a file specification must be entered. The device and directory should be the same as specified for the BINARY parameter.

**Qualifiers**

**/BEFORE=date-time**

Specifies the ending date for the data to be included in the report. The default is the first day of the current month. This, along with the default for the /SINCE qualifier, causes the report to be a summary of the previous month's data.

**/CPU**

Selects CPU capacity data to be included in the report. This is the default.

**/DISK**

Selects disk capacity data to be included in the report. This is the default.

**/ERRORLOG**

Selects errorlog data to be summarized in the report. This is the default.

**/IO**

Selects system I/O data to be included in the report. This is the default.

**/MAIL=(email-address[,…])**

Specifies one or more e-mail addresses to which the summary report will be mailed. By default, the report is not mailed.

**/MEMORY**

Selects memory capacity data to be included in the report. This is the default.

**/OUTPUT=SYS$OUTPUT**

Specifies the name of the output report file.

**/PAGE_SIZE=60**

Specifies the number of lines per page in the output report.

**/RENAME=.PROCESSED**

By default, OSM will rename all files that it processes to generate the report to the specified name. Only those files processed that do not include any data after the time specified by the /BEFORE qualifier will be renamed. This is used to make it easier to delete the files that will no longer be needed for the generation of future reports. To not have the files renamed, specify /NORENAME.

**/SECURITY**

Selects security audit data to be summarized in the report. This is the default.

**/SINCE=date-time**

Specifies the starting date for the data to be included in the report. The default is the first day of the previous month.

**/SYSTEM**

Selects general system data to be included in the report. This is the default.

## SUMMARIZE/ERRLOG

The SUMMARIZE/ERRLOG command processes binary OpenVMS errorlog files to create a summary report of the data contained in them.

**Format**

**SUMMARIZE/ERRLOG input-file**

**Parameters**

**input-file**

Specifies the name of the input file to be summarized. This file is a binary OpenVMS errorlog file. Only one file may be specified. Wildcard characters are not supported. There is no default, so a file specification must be entered.

**Qualifiers**

**/COUNT**

Specifies that a summary listing the number of each type of errorlog entry encountered be included in the output report.

**/DISK**

Selects disk error data to be included in the report. This is the default.

**/ERRORS**

Specifies that a summary listing of the device errors be included in the output report.

**/OTHER**

Selects error data other than disk and tape to be included in the report. This is the default.

**/OUTPUT=SYS$OUTPUT**

Specifies the name of the output report file.

**/TAPE**

Selects tape error data to be included in the report. This is the default.

**/TIME**

Displays time of earliest and latest errorlog entries in the report. This is the default.

# Summary Parameters

These parameters control what the header looks like in the summary reports. Specifying these parameters will allow you to assign bold, italic, or whatever other characteristics the output device supports to the headers in the output report file.

### ENHANCE=

This is normally a string which will cause the following text to be bolded (or whatever other characteristic the device supports) on the printer on which the report is being printed. The default for this parameter is a null string, indicating no special header formatting.

### NORMAL=

This parameter is used to disable whatever the OSM parameter ENHANCE has enabled. The default for this parameter is a null string, indicating no special header formatting.

# Example Summary Report

The following is an example summary report. Text is interspersed with the summary report to explain the different sections.

The first section of the report is the customer and OSM information. OSM gets this data from the customer contact parameters.

```
********************
OSM Summary V2.2                                                    Page 1
VMS V5.5-2  on node YOGI a VAXstation 2000
Report for  1-NOV-1996 00:00 until  1-DEC-1996 00:00
_____________________________________________________________________________

_____________________________________________________________________________

Customer Information
_____________________________________________________________________________

Customer:   The PARSEC Group
Contact:    Jim Mehlhop
Phone:      303-763-9600
Account:    8
_____________________________________________________________________________

_____________________________________________________________________________

OSM Information
_____________________________________________________________________________

OSM V3.0
Heartbeat:  3
Prime time is 8:00-17:00
_____________________________________________________________________________
```

This section of the report summarizes all the event records OSM has received over the period.

```
_____________________________________________________________________________

Report Summary
_____________________________________________________________________________

         17 Page and swap file usage records
          1 High cpu mode time records
         50 Memory pool usage records
         14 Logical name table records
          4 High page fault records
         12 High disk usage records
         10 File system cache records
          7 Error log records
         19 System capacity planning records
         58 Disk capacity planning records
        138 Informational records
         53 Files processed
_____________________________________________________________________________
```

This section displays performance related data summarized over the specified period.

```
_____________________________________________________________________________

General System Information
_____________________________________________________________________________

There were 4 reboot(s) detected.
There were 1 report(s) of high CPU mode activity.

Page file has 49992 blocks with a peak of 13788 used.
Swap file has 2296 blocks with a peak of 2160 used.

SRPCOUNT is 415 with a peak usage of 2048.
IRPCOUNT is 277 with a peak usage of 1093.
LRPCOUNT is 32 with a peak usage of 78.
NPAGEDYN is 481792 with a peak usage of 739840.

Effective LNMSHASHTBL is 128 with a maximum of 127 used.
There has been a maximum of 1096 logical names with 23 in one hash chain.
Average number of logicals in each hash chain used reached 8.

The system fault rate reached 3 per second 3 time(s).
It reached 2 faults per second or higher 4 time(s).
First errorlog entry is at 21-JUL-1996 10:34.
_____________________________________________________________________________
_____________________________________________________________________________

System I/O Information
_____________________________________________________________________________

There were 12 report(s) of high disk usage for _$1$DKA0:.

There were 9 report(s) referencing directory index cache (ACP_DINDXCACHE).
There were 1 report(s) referencing file header cache (ACP_HDRCACHE).
_____________________________________________________________________________
```

This section displays a summary of hardware exception events detected over the specified period.

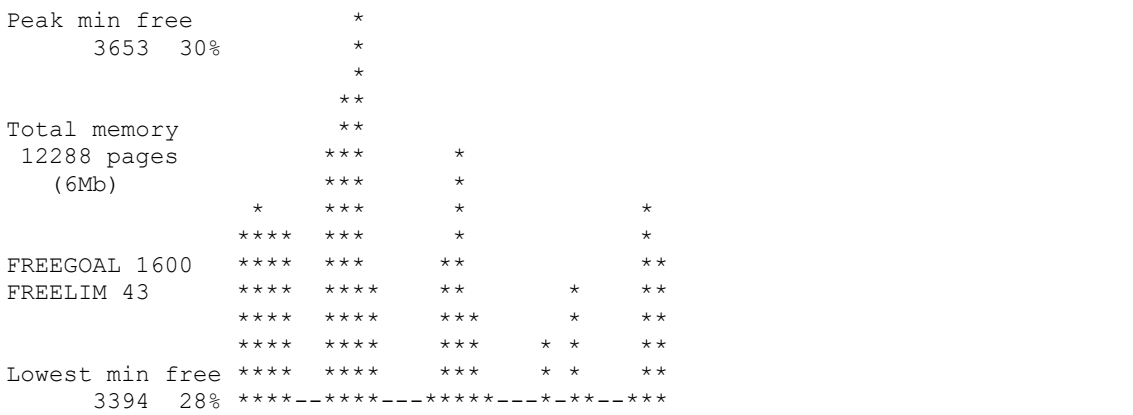
```
_____________________________________________________________________________

Errorlog Summary
_____________________________________________________________________________

 V A X / V M S        SYSTEM ERROR REPORT         COMPILED 21-DEC-1996 10:40:12
                                                                  PAGE   1.

 SUMMARY OF ALL ENTRIES LOGGED BY SID 08000000
       VOLUME MOUNT                          4.
       VOLUME DISMOUNT                       3.
       DATE OF EARLIEST ENTRY          8-NOV-1996 11:21:31.91
       DATE OF LATEST ENTRY           23-NOV-1996 10:09:56.58
_____________________________________________________________________________
```

This section graphically displays the minimum amount of free memory observed over the specified period of prime time. The dashes (-) indicate that OSM did not collect capacity data over that period of time. Capacity data collection is controlled by the OSM parameter PRIME which; by default, does not include Saturday and Sunday. Another reason data may not be available would be that OSM was not running at the time.

The graph's lower limit scale is determined by the minimum amount of free memory observed over the specified prime period. The graph's upper limit is determined by the highest minimum amount of free memory observed over that same period.

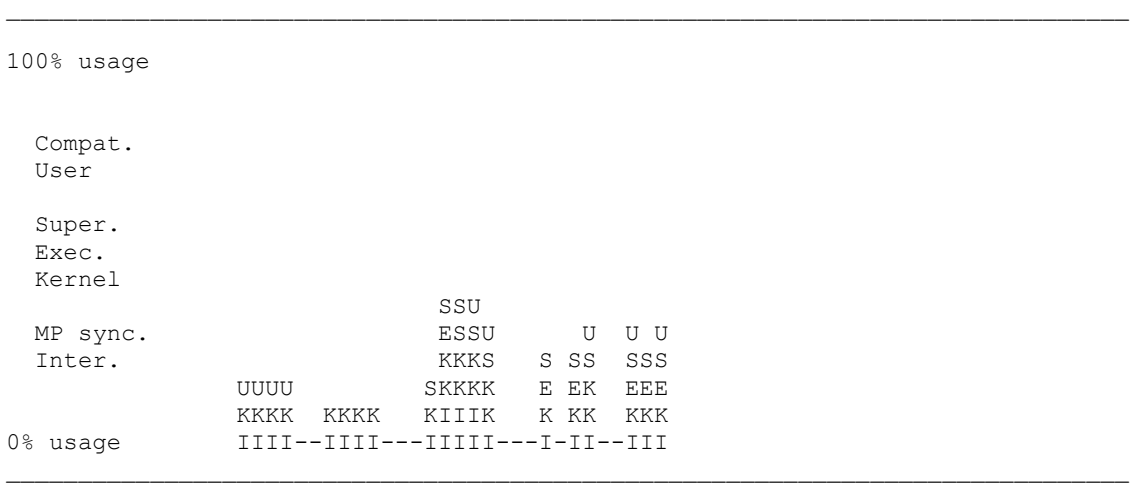
In this example, the minimum amount of free memory observed on one day was 3,653 pages and on another 3,394 pages. Every other day the minimum amount of free memory was in between those two values.

```
Memory Capacity Summary (- no data available)
```




This section graphically displays the amount of time spent in each of the CPU access modes observed over the specified period of prime time. The dashes (-) indicate that OSM did not collect capacity data over that period of time. Capacity data collection is controlled by the OSM parameter PRIME. Another reason data may not be available would be that OSM was not running at the time.

```
CPU Capacity Summary (- no data available)
```

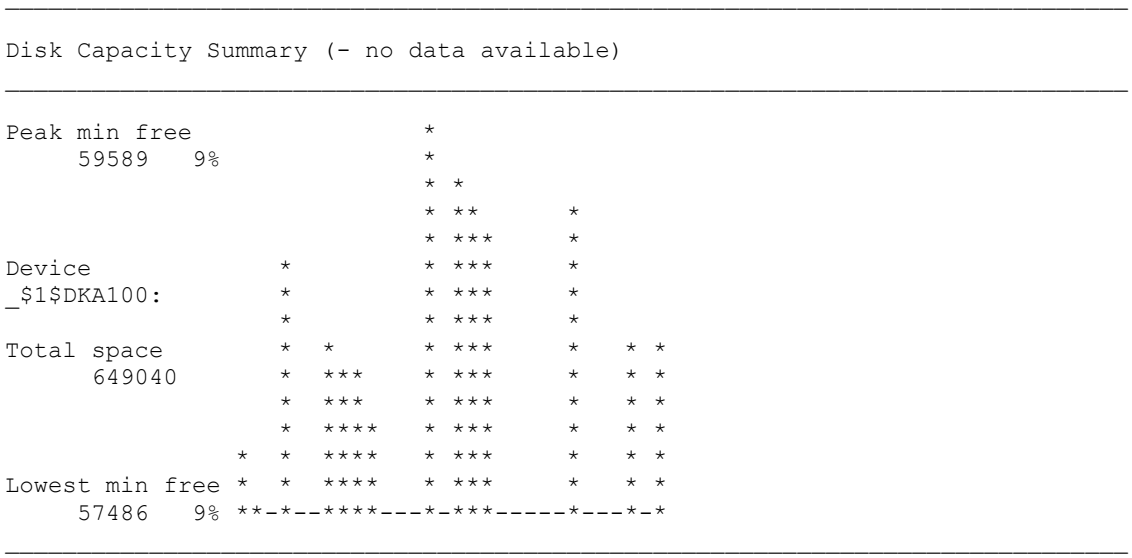


This section graphically displays the minimum amount of free disk space for each disk observed over the specified period of prime time. The dashes (-) indicate that OSM did not collect capacity data over that period of time. Capacity data collection is controlled by the OSM parameter PRIME. Another reason data may not be available would be that OSM was not running at the time.

The graph's lower limit is determined by the minimum amount of free disk space observed over the specified prime period. The graph's upper limit is determined by the highest minimum amount of free disk space observed over that same period.

In this example, the minimum amount of free disk space observed on one day was 59,589 blocks and on another 57,486 blocks. Every other day the minimum amount of free disk space was in-between those two values.

There will be a graph for each disk on the system.



# Chapter 15
# OSM Installation

## Overview

On OpenVMS, the standard VMSINSTAL command procedure located in the SYS$UPDATE directory is used to install and update OSM. In order to use this product, you must install a license code key using the OpenVMS License Management Facility (LMF). When you are ready to install OSM call PARSEC Group at 303-763-9600 and you will be supplied with the required license code via phone, FAX, or E-mail.

### Installing OSM in a VMScluster Environment

In a VMScluster environment, it is only necessary to run the VMSINSTAL command procedure once. After doing so on the first node in the cluster, log into other nodes and execute the following command:

```
$ @disk:[OSM]OSM_ADDNODE
```

In this command, disk refers to the disk on which OSM has been installed.

## Updating OSM

During an update to OSM, the installation procedure will stop the monitor and the transport processes if they are currently running. If you elect not to have the installation procedure start OSM automatically, you may start it later with the following command:

```
$ @SYS$MANAGER:OSM_STARTUP
```

# OpenVMS OSM Installation Example

The following example shows a complete OSM installation procedure as done on an HP Integrity rx2600 running OpenVMS V8.4. Your responses and the output you receive will vary based on your configuration.

1. To begin, verify that you are logged into the SYSTEM account. The standard OpenVMS installation procedure is designed to work with the OpenVMS-supplied parameters for the SYSTEM account. It is recommended (not required, as the example shows) that this account be used for all software installations. If you do not have the required privileges or quotas, you will receive a warning and the installation procedure may fail.
2. OSM utilizes the License Management Facility provided with OpenVMS. Use SYS$UPDATE:VMSLICENSE.COM to install it. If you have received your license from PARSEC Group, then install it at this time. If you have not received your license key, please contact your sales representative at:

   PARSEC Group
   5610 Ward Rd, Suite 100
   Arvada, CO 80002
   Telephone 303-763-9600
   FAX 303-763-9909

3. Invoke the installation procedure. You may enter the full command by specifying the product and device, or you will be prompted for that information.

```
$ @sys$update:vmsinstal osm042 dsa2:[software]


         OpenVMS  Software Product Installation Procedure V8.4


It is 27-FEB-2024 at 06:51.

Enter a question mark (?) at any time for help.
```

4. VMSINSTAL checks for other users on the system, verifies that you are logged into the SYSTEM account and verifies that the process quotas are sufficient to do a software installation. If not, it outputs the appropriate messages and asks if you wish to continue. If any of the process quotas are too low, do not continue the installation of OSM. Exit the installation procedure, change the quotas in authorize, log off and back on, then restart the installation procedure. Since you do not need to reboot the system to install OSM, other users can stay logged on during the installation. Enter Y, then press <RETURN> to continue, or press <RETURN> to stop the installation and return to the $ prompt.

```
%VMSINSTAL-W-NOTSYSTEM, You are not logged in to the SYSTEM account.
%VMSINSTAL-W-ACTIVE, The following processes are still active:
       WILLIAMS
* Do you want to continue anyway [NO]? y
```

5. VMSINSTAL asks whether you have recently backed up your system disk. It is always a good idea to have a current backup of your system disk before installing any software. If you wish to backup your system disk before continuing with the installation of OSM, press N followed by <RETURN> and VMSINSTAL will exit the installation. After completing the backup, restart this installation from step 1. To continue with the installation, press <RETURN>.

```
* Are you satisfied with the backup of your system disk [YES]?
```

6. On OpenVMS V8.4 and later VMSINSTAL checks if the software was signed by HP and if that signature is valid. OSM is provided by PARSEC Group and is not signed by HP, so answer yes if this question is asked.

```
The following products will be processed:

  OSM V4.2


        Beginning installation of OSM V4.2 at 06:51

    ******************************************************************
%VMSINSTAL-I-VALSIGN, Performing product kit validation of signed kits …
%VMSINSTAL-I-NOVALDONE, Product is not signed by HP
* Do you want to install this product [NO]? y
```

7. VMSINSTAL now begins the installation of OSM.

```
%VMSINSTAL-I-RESTORE, Restoring product save set A …

Installation of OSM V4.2
```

8. The procedure reminds you of the phone number to call if you have any questions or problems regarding this or any other product supplied by PARSEC Group.

```
Copyright (c) 2024 by The PARSEC Group Inc., all rights reserved

Thank you for choosing OSM as your one and only on-line system manager
tool. You will be glad you did!

You, the person performing this installation, will probably be in a
position to know the most about how the product is used at your site.
If you think of any way that we can improve OSM to better meet your
needs and the needs of others, we would appreciate your call.  The
PARSEC Group products contain features which were originally suggested
by our customers.

Call us at (303) 763-9600 and ask for Technical Support if we can assist
you during the installation or at any other time while you are running OSM.

Press <RETURN> to continue.
```

9. OSM is licensed for use on the system(s) for which it was purchased or provided as a demo. The installation procedure reminds you that this software should only be installed on system(s) for which it is licensed. Press <RETURN> to continue the installation.

```
 Proprietary Rights Notice:

 All rights reserved.  This software contains valuable properties
 and trade secrets of The PARSEC Group.  This software may not be
 reproduced or transmitted in any form or by any means, in part or
 in whole, without permission from:

     PARSEC Group
     5610 Ward Road
     Suite 110
     Arvada, CO  80002
     (303) 763-9600

 It is illegal to install or run OSM on any system that is not
 licensed.  If you are not sure which systems your company has
 licensed, check with The PARSEC Group before proceeding.  Demo
 sites may install and run OSM for the period specified by their
 demo license.

* Have you carefully read all of the above notice [YES]?
```

10. OSM is licensed using the OpenVMS License Management Facility (LMF). You may purchase a license by contacting your sales representative at PARSEC Group or sending an email to experts@parsec.com. You may request a 30-day demo license to try out the software before purchasing. If the installation is not able to detect a loaded license for OSM, then this question may be asked.

```
OSM requires a license to function.  Once you have received a license,
it may be installed using @SYS$UPDATE:VMSLICENSE.COM.
* Have you received and installed your license for OSM [YES]?
```

11. The installation procedure outputs a message indicating that it will create a directory named [OSM] to contain the applications files. OSM requires a relatively small amount of disk space for its installation and execution. It creates a system logical name OSM$ which points to that directory. It will prompt you for the name of the device that you want the OSM directory and files to reside on.

    If you wish to run the product on multiple nodes in a cluster, install the product on a disk available to all nodes.

    If this is an upgrade, and the logical OSM$ is defined, then the installation notifies you that it will update the existing directory to the version being installed. You will be asked if you want to purge files replaced by this installation.

```
Now you must determine on which DISK to put the new OSM files. OSM V4.2
creates a logical name of OSM$ which points to the directory on that
disk containing these files. The directory name is [osm].  If this
directory already exists, the new files will be copied over any
existing files in that directory.  The basic OSM application
uses less than 20000 blocks.

* Enter the name of the DEVICE to put OSM on: sys$sysdevice:
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
DISK$CLASS3I64SYS:[OSM].
```

12. The OSM installation procedure allows the product to be started automatically. The default is YES, so if you want OSM to start automatically enter <RETURN>. If you answer NO then the installation procedure will continue but will not start OSM.

```
* Do you want OSM started during installation [YES]?
```

13. If you are in a VMScluster you are prompted to identify the node(s) on which node(s) the transport process may run. The transport process processes the binary event files for all the nodes in the cluster. You may specify multiple nodes where it may run by separating them with commas. The default answer is the name of the node on which you are installing the product.

```
You may specify on which nodes OSM will run the transport process.
To specify more than one destination, enter all desired nodes
separated by commas.

* Which node(s) of the cluster will be the transport node(s) [CLASS3]:
```

14. The OSM installation procedure then asks for an account number, which will be placed in the OSM parameter file. The account number will help you identify which nodes are generating messages. For more information on the account number, refer to the OSM parameter ACCOUNT. In our example we are using account 9999, your value could be different.

```
* What will your account number be on this system: 9999
```

15. The following prompts simply help to identify this system and the individual responsible for its administration.

```
The next five questions ask for the company name, up
to three lines of address information, and the contact
name.  You may leave any or all of these lines blank
by pressing only the RETURN key.

* Company name: PARSEC Group
* First line of address: 5610 Ward Rd.
* Second line of address: Suite 100
* Third line of address: Arvada, CO 80002
* Contact name: David Jones
Please enter the phone number, including area code, of the individual
who should be contacted regarding any problems discovered on this
system by OSM.
* Contact phone number: (303) 763-9600
```

16. This section of the installation procedure refers to the reporting capabilities of OSM. It asks you how you would like OSM to report exception event information. The most common of these is via mail, which we selected in this example. Since we specified mail, the installation procedure prompts for more information about the user to receive the mail. For more information on these options, refer to *Chapter 9: Exception Event Reporting* and *Chapter 10: OSM Paging Support* in this manual.

```
OSM is capable of sending its output to several different locations.
You will be requested to enter your choice below.  To specify more
than one destination, enter all desired destinations, separated by
commas. The options are:

    BINARY   - a copy of the binary file in another
               directory or node
    ERRORLOG - a copy of errorlog information in
               a separate binary file
    FILE     - an ASCII printable file
    MAIL     - an OpenVMS mail message
    OPCOM    - a message to operator enabled terminals
    PAGER    - page a cell phone or PDA via E-Mail

* Enter the destination of OSM information from above: mail
* To which users should the mail be sent: jones,djones@parsec.com
```

If performing an upgrade, the defaults come from the current parameter files.

17. The installation checks for the existence of a swap file. Some systems do not have swap files. When OSM runs, it considers the lack of a swap file a critical situation and reports it. To prevent this, the installation procedure disables swap file checking if you indicate that the absence of a swap file is normal.

```
There is currently no swap file installed.
* Is this normal for this system? y
```

18. From this point, the installation procedure continues without asking any more questions. It will remind you that there are several template files that can be tailored to your needs and incorporated into OSM. The template files are well documented, and it is strongly recommended that you look at them if you are trying to implement the specified OSM feature.

```
%OSM-I-DONEASK, No further questions will be asked during this installation.
OSM provides to the user some template files which may be used as
examples during the setup of OSM.  They are located in the OSM$
directory and are listed here for your information.

Example parameter file which can be used by OSM
    OSM_CTLFILE.TEMPLATE

Example command procedure to define OSM logical names
    OSM_LOGICALS.TEMPLATE               OSM_SYSUAF_LOGICALS.TEMPLATE

Example command procedures for user defined monitoring
    CHECK_ENTRY_UDM.TEMPLATE            CHECK_LICENSE_UDM.TEMPLATE
    CHECK_PRINT_QUEUE_UDM.TEMPLATE      DS100_DS200_MONITOR_UDM.TEMPLATE
    DS90LPLUS_MONITOR_UDM.TEMPLATE      DSSI_MONITOR_UDM.TEMPLATE
    EXTRACT_NEWMAIL_UDM.TEMPLATE        OSM_SUMMARY_UDM.TEMPLATE

Example automated fix command procedures
    DISK_PURGE_FIX.TEMPLATE             HIGH_KERNEL_TIME_FIX.TEMPLATE
    PAGE_SWAP_FILE_FIX.TEMPLATE         T4_FIX.TEMPLATE
%OSM-I-PROGRESS, Removing old files which are no longer necessary.
%OSM-I-PROGRESS, Building executables.
Copyright (c) 1994 - 2024 The PARSEC Group Inc.  All rights reserved.
%OSM-I-PROGRESS, Setting up new version.
```

19. The next portion of the procedure reminds you that making the product available and starting it with default values requires that you add one line to the site-specific startup command procedure and one line to the system-wide login command procedure.

```
The installation has completed.  It has created the logical name OSM$
which points to the Disk and Directory that contains the utility. It has
also created a foreign command named OSM to activate the utility. In order
for the logical name and the command to be defined upon reboot or login
insert the next 2 lines in 2 different system command procedures.

Insert this line in SYS$MANAGER:SYSTARTUP_VMS.COM
 $ @SYS$MANAGER:OSM_STARTUP

Insert this line in SYS$MANAGER:SYSHUTDWN.COM
 $ @SYS$MANAGER:OSM_SHUTDOWN

Insert the following line in the system wide login
command file:  SYS$SYSROOT:[SYSMGR]SYLOGIN.com;
 $ IF F$TRNLNM("OSM$") .NES. "" THEN $OSM :== $OSM_CONTROL
```

20. If you have installed OSM in a VMScluster environment the following text is displayed.

```
Since you have just installed OSM in a VMScluster environment, if you
are running from a COMMON system disk, you can configure the other nodes
on this disk. This is accomplished by logging into each of the other
systems and executing the following command:
 $ @DISK$CLASS3I64SYS:[OSM]OSM_ADDNODE

You will need a separate account number and license for each node.
Press <RETURN> to continue.
```

21. The installation now informs you of the steps that are needed if you wish to have OSM automatically send crash data upon detection of a system crash to the specified location(s).

```
OSM has the capability to collect and send partial system crash data.
In order for this to work you must have a valid system dump file.
During system startup this dump file can be examined to collect
information.  If you are already doing this during startup, modify the
procedure that does this, and add the line @SYS$MANAGER:OSM_SDA_COMMANDS.
If you are not already examining this dump file then add the line
@SYS$MANAGER:OSM_SDA in the system startup file prior to the startup of OSM.
```

22. At this point, the installation procedure moves the files to their target directories and OSM is started if requested earlier.

```
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories…
Copyright (c) 1994 - 2024 The PARSEC Group Inc.  All rights reserved.

%OSM-I-STARTUP, OSM is starting up
OSM started with pid 20C0069B

        Installation of OSM V4.2 completed at 06:53

    Adding history entry in VMI$ROOT:[SYSUPD]VMSINSTAL.HISTORY

    Creating installation data file: VMI$ROOT:[SYSUPD]OSM042.VMI_DATA


        VMSINSTAL procedure done at 06:54
```

23. If you chose not to start OSM during the install, you may at this time edit either the node-specific or cluster-wide OSM parameter file, depending on your environment. Both parameter files are in the OSM$: directory. The cluster-wide parameter file is named OSM_CTLFILE.DAT. The node-specific parameter file is named OSM_CTLFILE_*node*.DAT, where *'node'* is the system's node name. Below is an example of a setup parameter file created by the previous install procedure.

```
PARSEC Group
5610 Ward Rd.
Suite 100
Arvada, CO 80002
David Jones
phone=(303) 763-9600
output=MAIL
mail=JONES,DJONES@PARSEC.COM
```

    After adding any site-specific parameters required for the node, exit the editor. During the initial installation, you probably will not modify any other parameters.

    For a more detailed example of an OSM parameter file, type or print the file named OSM_CTLFILE.TEMPLATE in the OSM$ directory.

24. You are now ready to start the OSM processes if you did not do this during the installation. If you are not currently logged into an account with sufficient privileges to start OSM, do so now and then execute the following command:

```
$ @sys$manager:osm_startup
Copyright (c) 1994 - 2024 The PARSEC Group Inc.  All rights reserved.

%OSM-I-STARTUP, OSM is starting up
OSM started with pid 20C0069F
```

# Installing OSM on Additional VMScluster Node(s)

As stated in step 24 of the Installation Guide, if you are in a VMScluster, just log into each of the other nodes and execute the command procedure OSM_ADDNODE to enable OSM on that node in the VMScluster.

# Updating OSM on Additional OpenVMS DECnet Node(s)

Once OSM has been installed on an OpenVMS node, other nodes in the network that are running previous versions of OSM may be updated by executing OSM_UPDATE.COM. Enter the command @OSM$:OSM_UPDATE XXX where XXX is the DECnet name of the remote node. If this command procedure is not on the local system, then you may type @XXX::OSM$:OSM_UPDATE. For this to work correctly, you must have access to the file in the OSM$ directory on the remote node. This may be done either through the default DECnet account, if it is enabled, or through the use of proxies.

# License Installation Information

OSM utilizes the OpenVMS License Management Facility (LMF). These licenses are installed using the LICENSE REGISTER command, or the OpenVMS supplied procedure SYS$UPDATE:VMSLICENSE.COM.

# Appendix A
# Parameter List

The following is a list of all parameters used by OSM along with their default values. The chapter reference is to the chapter in which the parameter is documented. Any parameter that does not list a default value either does not have a default value or gets its default from other parameters or the current environment. Please see the referenced chapter for additional information. Room has been left after the default for most of the parameters to make notes that may be pertinent to your site.

| Parameter | Chapter | Default |
|---|---|---|
| ACCOUNT | 4 | (no default) |
| ACTIVE | 9 | 0:00-0:00,Sun,Mon,Tue,Wed,Thu,Fri,Sat |
| ADDRESS | 4 | (no default) |
| ANALYZE/ERROR | 9 | (see text) |
| AUDIT | 6 | Y |
| AUDIT_BREAKIN | 6 | 6 |
| AUDIT_BREAKIN_F | 11 | (no default) |
| AUDIT_BREAKIN_P | 10 | (no default) |
| AUDIT_BYPASS | 6 | 5 |
| AUDIT_BYPASS_F | 11 | (no default) |
| AUDIT_BYPASS_P | 10 | (no default) |
| AUDIT_INSTALL | 6 | 3 |
| AUDIT_INSTALL_F | 11 | (no default) |
| AUDIT_INSTALL_P | 10 | (no default) |
| AUDIT_LOGFAIL | 6 | 3 |
| AUDIT_LOGFAIL_F | 11 | (no default) |
| AUDIT_LOGFAIL_P | 10 | (no default) |
| AUDIT_READALL | 6 | 4 |
| AUDIT_READALL_F | 11 | (no default) |
| AUDIT_READALL_P | 10 | (no default) |
| AUDIT_UAF | 6 | 5 |

| Parameter | Chapter | Default |
|---|---|---|
| AUDIT_UAF_F | 11 | (no default) |
| AUDIT_UAF_P | 10 | (no default) |
| AUTOSTART | 5 | 300,TRANSPORT |
| BINARY | 9 | (no default) |
| BUGCHECK | 6 | 9 |
| BUGCHECK_F | 11 | (no default) |
| BUGCHECK_P | 10 | (no default) |
| BURST_DELAY | 5 | 15 |
| CAPACITY | 6 | 1 |
| CAPACITY_F | 11 | (no default) |
| CAPACITY_P | 10 | (no default) |
| COMMAND# | 7 | (no default) |
| COMMAND#_F | 11 | (no default) |
| COMMAND#_P | 10 | (no default) |
| CONFIGURE | 6 | 300,Y |
| CONTACT NAME | 4 | (no default) |
| CPU | 6 | 1,20,35,10 |
| CPU_F | 11 | (no default) |
| CPU_P | 10 | (no default) |
| CPU_WAKE | 5 | 300 |
| CRITICAL | 5 | 6 |
| DEADLOCK | 6 | 4,1,2 |
| DEADLOCK_C | 6 | 6,2,20 |
| DEADLOCK_CF | 11 | (no default) |
| DEADLOCK_F | 11 | (no default) |
| DEADLOCK_P | 10 | (no default) |
| DECNET_OBJECT | 8 | (no default) |
| DEFAULT_NETWORK | 8 | (see text) |
| DETAIL | 9 | SEVERITY |
| DETAIL_ERRORLOG | 9 | ALL |
| DIAGNOSE | 9 | /SUMMARY |
| DISABLED | 9 | (no default) |
| DISK | 6 | 2,95,200 |
| DISK_C | 6 | 7,99,32000 |

| Parameter | Chapter | Default |
|---|---|---|
| DISK_CF | 11 | (no default) |
| DISK_F | 11 | (no default) |
| DISK_P | 10 | (no default) |
| EDITOR | 5 | TPU |
| ELV | 9 | /SUMMARY |
| ENABLED | 9 | (no default) |
| ENHANCE | 14 | (no default) |
| ERRFMT | 6 | Y |
| ERRFMT_BUGCHECK | 6 | 8 |
| ERRFMT_BUGCHECK_F | 11 | (no default) |
| ERRFMT_BUGCHECK_P | 10 | (no default) |
| ERRFMT_BUS | 6 | 6 |
| ERRFMT_BUS_F | 11 | (no default) |
| ERRFMT_BUS_P | 10 | (no default) |
| ERRFMT_CARD | 6 | 1 |
| ERRFMT_CARD_F | 11 | (no default) |
| ERRFMT_CARD_P | 10 | (no default) |
| ERRFMT_DISK | 6 | 6 |
| ERRFMT_DISK_F | 11 | (no default) |
| ERRFMT_DISK_P | 10 | (no default) |
| ERRFMT_HARD_MEMORY | 6 | 6 |
| ERRFMT_HARD_MEMORY_F | 11 | (no default) |
| ERRFMT_HARD_MEMORY_P | 10 | (no default) |
| ERRFMT_INFORMATION | 6 | 0 |
| ERRFMT_INFORMATION_F | 11 | (no default) |
| ERRFMT_INFORMATION_P | 10 | (no default) |
| ERRFMT_LINE_PRINTER | 6 | 0 |
| ERRFMT_LINE_PRINTER_F | 11 | (no default) |
| ERRFMT_LINE_PRINTER_P | 10 | (no default) |
| ERRFMT_MAILBOX | 6 | 1 |
| ERRFMT_MAILBOX_F | 11 | (no default) |
| ERRFMT_MAILBOX_P | 10 | (no default) |
| ERRFMT_MEMORY | 6 | 4 |
| ERRFMT_MEMORY_F | 11 | (no default) |

| Parameter | Chapter | Default |
|---|---|---|
| ERRFMT_MEMORY_P | 10 | (no default) |
| ERRFMT_MISCELLANEOUS | 6 | 1 |
| ERRFMT_MISCELLANEOUS_F | 11 | (no default) |
| ERRFMT_MISCELLANEOUS_P | 10 | (no default) |
| ERRFMT_POWER | 6 | 5 |
| ERRFMT_POWER_F | 11 | (no default) |
| ERRFMT_POWER_P | 10 | (no default) |
| ERRFMT_REALTIME | 6 | 7 |
| ERRFMT_REALTIME_F | 11 | (no default) |
| ERRFMT_REALTIME_P | 10 | (no default) |
| ERRFMT_SYNCHRONOUS_ COMMUNICATION | 6 | 4 |
| ERRFMT_SYNCHRONOUS_ COMMUNICATION_F | 11 | (no default) |
| ERRFMT_SYNCHRONOUS_ COMMUNICATION_P | 10 | (no default) |
| ERRFMT_SYSTEM | 6 | 6 |
| ERRFMT_SYSTEM_F | 11 | (no default) |
| ERRFMT_SYSTEM_P | 10 | (no default) |
| ERRFMT_TAPE | 6 | 2 |
| ERRFMT_TAPE_F | 11 | (no default) |
| ERRFMT_TAPE_P | 10 | (no default) |
| ERRFMT_TERMINAL | 6 | 0 |
| ERRFMT_TERMINAL_F | 11 | (no default) |
| ERRFMT_TERMINAL_P | 10 | (no default) |
| ERRFMT_VOLUME | 6 | 1 |
| ERRFMT_VOLUME_F | 11 | (no default) |
| ERRFMT_VOLUME_P | 10 | (no default) |
| ERRFMT_WORKSTATION | 6 | 4 |
| ERRFMT_WORKSTATION_F | 11 | (no default) |
| ERRFMT_WORKSTATION_P | 10 | (no default) |
| ERRORLOG | 9 | OSM$TPG:%04d%02d%02d%02d.DAT |
| ERRORLOG_FORMAT | 9 | OSM |
| ERRORLOG_TRANSLATE | 9 | MONITOR |
| EXIT | 3 | (no default) |

| Parameter | Chapter | Default |
|---|---|---|
| FAX | 4 | (no default) |
| FILE | 9 | OSM$:.OSM_ASCII; |
| GBLSECTIONS | 6 | 4,90,80 |
| GBLSECTIONS_C | 6 | 6,97,95 |
| GBLSECTIONS_CF | 11 | (no default) |
| GBLSECTIONS_F | 11 | (no default) |
| GBLSECTIONS_P | 10 | (no default) |
| HBOUTPUT | 9 | MAIL |
| HEARTBEAT | 5 | 0 |
| IO | 6 | 3,20,10 |
| IO_C | 6 | 8,32000,32000 |
| IO_CF | 11 | (no default) |
| IO_F | 11 | (no default) |
| IO_P | 10 | (no default) |
| LNM_TABLE | 5 | LNM$FILE_DEV |
| LOG | 5 | (no default) |
| LOGICAL_NAMES | 6 | 1,65,25,6 |
| LOGICAL_NAMES_F | 11 | (no default) |
| LOGICAL_NAMES_P | 10 | (no default) |
| MAIL | 9 | (no default) |
| MONITOR_WAKE | 5 | 30 |
| NETWORK# | 8 | (no default) |
| NETWORK#_F | 11 | (no default) |
| NETWORK#_P | 10 | (no default) |
| NETWORK_ASCII | 8 | 0 |
| NETWORK_ASCII_F | 11 | (no default) |
| NETWORK_ASCII_P | 10 | (no default) |
| NETWORK_ID | 8 | (no default) |
| NETWORK_TIME# | 8 | (no default) |
| NETWORK_TIME#_F | 11 | (no default) |
| NETWORK_TIME#_P | 10 | (no default) |
| NETWORK_TIME_SET# | 8 | (no default) |
| NIGHT | 5 | 00:30-05:00 |
| NORMAL | 14 | (no default) |

| Parameter | Chapter | Default |
|---|---|---|
| OPCOM | 5 | N |
| OPCOM_ERROR | 5 | 1 |
| OPCOM_ERROR_P | 10 | (no default) |
| OPERATOR | 9 | CENTRAL |
| ORGANIZATION NAME | 4 | (no default) |
| OUTPUT | 9 | MAIL |
| OUTPUT_ACCOUNTS | 9 | (no default) |
| OUTPUT_GROUP | 9 | 0,0 |
| OUTPUT_REASONS | 9 | ALL |
| OUTPUT_SEVERITY | 9 | 0 |
| OUTPUT_SEVERITY_MAX | 9 | 10 |
| PAGEFILE | 6 | 3,65,70 |
| PAGEFILE_C | 6 | 8,85,90 |
| PAGEFILE_CF | 11 | (no default) |
| PAGEFILE_F | 11 | (no default) |
| PAGEFILE_P | 10 | (no default) |
| PAGER | 10 | 0 |
| PAGER_BACKUP | 10 | (no default) |
| PAGER_CHAR | 10 | (see text) |
| PAGER_DATA_BITS | 10 | 0 |
| PAGER_DELETE | 10 | 14 0:0:0 |
| PAGER_RETRY | 10 | 1,0 00:00:00 |
| PAGE_FAULTS | 6 | 2,500,40,2 |
| PAGE_FAULTS_F | 11 | (no default) |
| PAGE_FAULTS_P | 10 | (no default) |
| PHONE | 4 | (no default) |
| POOL | 6 | 3,30,65 |
| POOL_C | 6 | 8,85,90 |
| POOL_CF | 11 | (no default) |
| POOL_F | 11 | (no default) |
| POOL_P | 10 | (no default) |
| POWER | 6 | 5 |
| POWER_F | 11 | (no default) |
| POWER_P | 10 | (no default) |

| Parameter | Chapter | Default |
|---|---|---|
| PRIME | 5 | 8:00-17:00,Mon,Tue,Wed,Thu,Fri |
| PRINTER | 9 | (no default) |
| PROCESSES | 6 | 5,80,85 |
| PROCESSES_C | 6 | 8,90,95 |
| PURGE_WORKING_SET | 5 | N |
| REPEAT_ALWAYS | 9 | N |
| REPEAT_DOM | 9 | 0 |
| REPEAT_DOW | 9 | 0 |
| SITE | 4 | (no default) |
| SYSUAF | 12 | (no default) |
| SYSUAF_ALLOW | 12 | (no default) |
| SYSUAF_DISTRIBUTE | 12 | (no default) |
| SYSUAF_DISTRIBUTE_ADD | 12 | (no default) |
| SYSUAF_DISTRIBUTE_VERIFY | 12 | (no default) |
| SYSUAF_DISTRIBUTE_VERIFY_ADD | 12 | (no default) |
| SYSUAF_NODE | 12 | (see text) |
| SYSUAF_NOREMOVE | 12 | SYSTEM,DEFAULT |
| SYSUAF_NOTIFY | 12 | (no default) |
| TCPIP_PORT | 8 | 0,0 |
| TEMPERATURE | 6 | (no default) |
| TEMPERATURE_C | 6 | (no default) |
| TEXT | 4 | (no default) |
| TEXT_ADD | 4 | (no default) |
| TEXT_FORMAT | 9 | BRIEF |
| TEXT_MODE | 9 | FILE |
| TEXT_POSTFIX | 9 | (no default) |
| TEXT_POSTFIX_ADD | 9 | (no default) |
| TEXT_PREFIX | 9 | (no default) |
| TEXT_PREFIX_ADD | 9 | (no default) |
| TIME_SET | 8 | 300,0 0:00:15 |
| TRANSPORT_WAKE | 5 | 300 |
| UIC | 5 | [1,4] |
| XQP_CACHE | 6 | 3,4,90 |
| XQP_CACHE_F | 11 | (no default) |

| Parameter | Chapter | Default |
|---|---|---|
| XQP_CACHE_P | 10 | (no default) |

# Appendix B
# OSM File Summary

## PERMANENT FILES

Unless otherwise noted all files will be in the main OSM directory. This is pointed to by the logical name OSM$. Several of the files include an architecture type in the file name. VAX is for OpenVMS VAX, ALPHA and AXP are for OpenVMS Alpha, and I64 is for OpenVMS for Integrity Servers.

**aaaaaa.DIR** (Directory file)

This directory is created, if necessary, during the startup of OSM. The aaaaaa in the directory name represents the node name for the node on which it is used. OSM$LOCAL is a logical name that points to this directory.

**BINARY.DIR** (Directory file)

This subdirectory is created in the OSM$ directory by the OSM_SUMMARY_SETUP.COM procedure. This directory is used to store a copy of the event files for later summarization. The BINARY parameter will point to this directory.

**ERRLOG_ALPHA.SYS** (Errorlog Files)
**ERRLOG_I64.SYS** (Errorlog Files)
**ERRLOG_VAX.SYS** (Errorlog Files)

These errorlog files may be used by the DEMONSTRATE ERRORLOG command for OSM to copy ERRFMT information to the monitor process.

**LINK.COM** (DCL Command procedure)

This is the command procedure that links all of the OSM executables. You can execute this command procedure to create new executables. This is normally performed after you have updated your system to a new major release of VMS.

**OSM_AXP_yyyy.EXE** (OpenVMS Executable image)
**OSM_IA64_yyyy.EXE** (OpenVMS Executable image)
**OSM_VAX_yyyy.EXE** (OpenVMS Executable image)

These are the executable images that the OSM monitor process runs to monitor and collect information. The yyyy in the file name represents the version of OpenVMS used to link the executable. For example, on an Alpha running OpenVMS Alpha V7.3-2 the name of the executable would be OSM_AXP_0732.EXE.

**OSM.LOG** (Log file)

This is the OSM monitor processes log file. It will contain information on the execution of the OSM monitor process. There will be a new log file created every time you start up the OSM monitor process. This file will be purged to five versions by the OSM startup procedure.

**OSM.THRESHOLDS** (Text data file)

This file contains the thresholds that the OSM process compares device error counts against to see if there have been enough errors to create a critical message.

**OSM_ADDNODE.COM** (DCL Command procedure)

This command procedure is used to set up OSM on additional nodes in the VMScluster. This command procedure is executed after the initial OSM installation has been completed.

**OSM_CONTROL_AXP_yyyy.EXE** (OpenVMS Executable image)
**OSM_CONTROL_IA64_yyyy.EXE** (OpenVMS Executable image)
**OSM_CONTROL_VAX_yyyy.EXE** (OpenVMS Executable image)

These are the image files that are executed when you issue any OSM command such as OSM START MONITOR. The yyyy in the file name represents the version of OpenVMS used to link the executable. For example, on an Alpha running OpenVMS I64 V8.3, the name of this image would be OSM_CONTROL_IA64_0830.EXE.

**OSM_CTLFILE.DAT** (Text data file)

This file is the cluster-common control file that defines how the monitor and transport processes execute.

**OSM_CTLFILE_aaaaaa.DAT** (Text data file)

This is the node-specific control file. There will be one of these files for every node that has been configured by either the OSM installation procedure or OSM_ADDNODE.COM. Parameters in this file will override the same parameters found in the cluster-common control file. The account number must be stored in this file. The aaaaaa in the file name is the name of the node for which the file is associated. The node name must match the SYSGEN parameter SCSNODE if it is defined. If SCSNODE is not defined, then this must match the DECnet node name if DECnet is running. If DECnet is not running and the SCSNODE parameter is blank, then the name of the file is OSM_CTLFILE_.DAT.

**OSM_LIB.OLB** (Object library file)
**OSM_LIB_IA64.OLB** (Object library file)
**OSM_LIB_AXP.OLB** (Object library file)

These are the object libraries that contain the objects used to link the various executables included in the OSM application. OSM_LIB.OLB contains the OpenVMS VAX versions of the objects.

**OSM_LOGICALS.COM** (DCL Command procedure)

This file is created during the installation of OSM if it does not already exist. It is a copy of OSM_LOGICALS.TEMPLATE or OSM_LOGICALS.TPL. This file is executed during the startup of OSM to create site specific logical names or environment variables to be used by OSM.

**OSM_LOGICALS.TEMPLATE** (OpenVMS Template File)

This is a template file for OSM_LOGICALS.COM.

**OSM_GETDEVICE_INFO.COM** (DCL Command procedure)

This command procedure may be run during the installation of the product. It collects configuration information and stores it in the appropriate temporary files.

**OSM_HELP.HLB** (Help library)

This is the help library used by the HELP command within OSM.

**OSM_MON_START.COM** (DCL Command procedure)

This command procedure is executed to start the OSM monitor process.

**OSM_NODE_SETUP.DAT** (PC Binary data file)

This file is the setup information about all nodes for which OSM is to monitor or affect.

**OSM_INFO.DAT** (Binary data file)

This file contains information about OSM paging. OSM uses the information in this file to keep track of how often OSM has paged a pager and when the next page should be. By default, it also keeps the last two weeks' worth of paging history. The actual amount of paging history is determined by the OSM parameter PAGER_DELETE. If this file does not exist, OSM automatically creates it the next time the transport process is started.

**OSM_INFO.FDL**(RMS FDL file)

This is the FDL file used to create OSM_INFO.DAT. The installation procedure and the transport startup procedure use this file to recreate OSM_INFO.DAT if it does not exist.

**OSM_PARAMS.TRANSPORT** (Log file)
**OSM_PARAMS_*.MONITOR** (Log file)

This is the log file that contains the parameters used by the OSM monitor and transport processes the last time they were started. For the monitor process, the * in the file name is replaced with the name of the node on which the monitor is running. These files are displayed by the OSM command SHOW PARAMETERS.

**OSM_REMOVE.COM** (DCL Command procedure)

This command procedure may be used to remove most OSM files from the system and to provide the user with messages about which other files may need to be deleted or edited to complete the removal of OSM from the system.

**OSM_SDA.COM** (DCL Command procedure)

This command procedure, which is found in the SYS$COMMON:[SYSMGR] directory, contains DCL commands to collect crash information at system startup time and store in the temporary data file SYS$SYSTEM:OSM_CRASH.DAT. In order for this to work you must have a valid system dump file. During system startup this dump file can be examined to collect information. If you are already doing this during startup, modify the procedure that does this, and add the line @SYS$MANAGER:OSM_SDA_COMMANDS. If you are not already examining this dump file then add the line @SYS$MANAGER:OSM_SDA in the system startup file prior to the startup of OSM.

**OSM_SDA_COMMANDS.COM** (SDA Command procedure)

This command file, found in the SYS$COMMON:[SYSMGR] directory, contains the SDA commands to collect the required crash information and store them in the temporary file SYS$SYSTEM:OSM_CRASH.DAT.

**OSM_SHUTDOWN.COM** (DCL Command procedure)

This command procedure, found in SYS$COMMON:[SYSMGR], shuts down the OSM application. It should be added to the system's site-specific shutdown command files to be executed at system shutdown time so that OSM will automatically be properly shut down.

**OSM_STARTUP.COM** (DCL Command procedure)

This command procedure, found in SYS$COMMON:[SYSMGR], starts the OSM application. It must be added to the system's site-specific startup command files to be executed at system boot time so that OSM will automatically start at each system boot.

**OSM_STARTUP.TEMPLATE** (Command procedure)

This template file is used during the installation of OSM and by OSM_ADDNODE.COM to build the OSM_STARTUP.COM procedure.

**OSM_TRANSPORT_AXP_yyyy.EXE** (OpenVMS Executable image)
**OSM_TRANSPORT_IA64_yyyy.EXE** (OpenVMS Executable image)
**OSM_TRANSPORT_VAX_yyyy.EXE** (OpenVMS Executable image)

These are the images run by the OSM_TRANSPORT process to send data to all of the destinations defined in the OSM control file. The yyyy in the file name represents the version of OpenVMS used to link the executable. For example, on a VAX running OpenVMS VAX V7.3, this name of this executable image would be OSM_TRANSPORT_VAX_0730.EXE.

**OSM_TRANSPORT.LOG** (Log file)

This is the OSM_TRANSPORT process log file. It contains output information from the OSM transport process. There will be a new log file created every time you start up the OSM_TRANSPORT process. The startup procedure will purge this file down to the last 5 versions.

**OSM_UPDATE.COM** (Command procedure)

This command procedure may be used to update OSM from remote nodes in the network.

**OSM_XPORT_START.COM** (DCL Command procedure)

This command procedure is executed to start the OSM_TRANSPORT process.

**OUTPUT_*.TEMPLATE** (Template file format)

These are template files showing possible ways to set up output scripts within OSM.

**TRANSPORT.REPEAT** (Data file)

This binary file contains information necessary for OSM to know when to redisplay data for each node based on its account number and the REPEAT parameters.

**WORK** (Directory file)

This directory is created to hold temporary files during the execution of OSM. Several temporary files may be created in this directory.

***_UDM.TEMPLATE** (Template file format)

These are template files showing possible ways to set up user-defined event monitoring within OSM.

***_FIX.TEMPLATE** (Template file format)

These are template files showing possible ways to set up automated fixes within OSM.

## Temporary Files

The following files are temporary files to facilitate the operation of the OSM application. Most of them will reside in the OSM$ directory unless otherwise identified.

**OSM_COMMAND*.*** (Text data file)

These ASCII files are created in the OSM$LOCAL directory by the execution of user-defined event monitoring commands. These files are deleted after OSM includes the data in its binary file.

**OSM_CRASH.DAT** (Text data file)

This file is created in the SYS$SPECIFIC:[SYSEXE] directory by the execution of the commands in OSM_SDA_COMMANDS.COM at system boot time to collect information about the crash to be sent to the designated parties. The OSM monitor process will look for this file when started. If it's located, it will include this information in the OSM data file for that system. This will be critical information that will be sent the next time that the OSM transport process wakes up to all destinations defined in the OSM control files.

**OSM_DATAFILE_aaaaaa.ACTIV** (Temporary binary data file)

This is the data file that the OSM SEND command creates when executed without the /ALL qualifier but with the /NOW qualifier. It is then renamed to include the .SEND extension to be processed by the OSM transport process. The aaaaaa will be the name of the system, obtained from one of the following sources: SCSNODE (SYSGEN parameter), SYS$NODE (logical name defined by DECnet startup), or SITE (OSM parameter defined in one of the OSM control files).

**OSM_DATAFILE_aaaaaa.ACTIVE** (Binary data file)

This is the data file that the OSM monitor process stores information in until the monitor process identifies a critical parameter or the transport process renames the file. The aaaaaa will be the name of the system, obtained from one of the following sources: SCSNODE (SYSGEN parameter), SYS$NODE (logical name defined by DECnet startup), or SITE (OSM parameter defined in one of the OSM control files).

**OSM_DATAFILE_aaaaaa.SEND** (Binary data file)

This is the data file that the OSM monitor process last stored binary event information in. This file is a renamed .ACTIVE file when it is determined that the data needs to be delivered to the appropriate party. The OSM_TRANSPORT process looks for this file at every wake-up interval to send this file's contents to all appropriate destinations in the OSM_CTLFILE.DAT. The aaaaaa will be the name of the system, obtained from one of the following sources: SCSNODE (SYSGEN parameter), SYS$NODE (logical name defined by DECnet startup, or SITE (OSM parameter defined in one of the OSM control files).

**OSM_INFORMATION.DAT** (Text file)

This is the file created in the SYS$SPECIFIC:[SYSEXE] directory by the OSM SEND REQUEST command. This is free form text entered by whatever editor the user has selected.

**OSM_INFORMATION.DEVICE** (Text file)

This file is created in the SYS$SPECIFIC:[SYSEXE] directory at installation time by the installation command procedure. It contains the output from the DCL command SHOW DEVICE/FULL. This file, along with OSM_INFORMATION.SYSGEN, will give an accurate description of the system configuration.

**OSM_INFORMATION.SYSGEN** (Text file)

This file is created in the SYS$SPECIFIC:[SYSEXE] directory at installation time by the Installation command procedure. It contains the output from the SYSGEN command SHOW CONFIGURATION command to be sent to the specified parties to help define the site environment.

# Handling Temporary Files

During every CPU_WAKE interval, the OSM process will look in the SYS$SPECIFIC:[SYSEXE] directory for the existence of any of the temporary files listed above. If it finds one (or more) of these files, it will include the file(s) contents in the OSM_DATAFILE_aaaaaa.ACTIVE file and then rename the .ACTIVE file to a .SEND file. After the OSM process has copied the data from the original file(s), it will then delete the original file(s). The OSM transport process will send the information from all 3 of these files to the designated parties. The aaaaaa in the above file name is the name of the node on which OSM is running.

# Index


ACCOUNT Parameter, 4-3, 9-13
ACTIVE Parameter, 9-6
Address Parameter, 4-3
AGE_ACCOUNTING Parameter, 6-5
AGE_ERRORLOG Parameter, 6-5
AGE_OPERATOR Parameter, 6-5
ALIGNMENT_FAULTS Parameter, 6-6
ANALYZE/ERROR Parameter, 9-6, 9-16, 9-17
ANALYZE/ERROR Parameter, 9-17
ANALYZE/ERROR/ELV Parameter, 9-12
ASCII Detail, 9-3
ASCII Files
  Controlling Detail, 9-3
  Repeat Data, 9-2
AUDIT Parameter, 6-7
AUDIT Parameter, 12-18
AUDIT_BREAKIN Parameter, 6-7
AUDIT_BREAKIN_F Parameter, 11-8
AUDIT_BREAKIN_P Parameter, 10-15
AUDIT_BYPASS Parameter, 6-7
AUDIT_BYPASS_F Parameter, 11-8
AUDIT_BYPASS_P Parameter, 10-15
AUDIT_INSTALL Parameter, 6-7
AUDIT_INSTALL_F Parameter, 11-8
AUDIT_INSTALL_P Parameter, 10-15
AUDIT_LOGFAIL Parameter, 6-7
AUDIT_LOGFAIL_F Parameter, 11-8
AUDIT_LOGFAIL_P Parameter, 10-15
AUDIT_READALL Parameter, 6-7
AUDIT_READALL_F Parameter, 11-8
AUDIT_READALL_P Parameter, 10-15
AUDIT_UAF Parameter, 6-7
AUDIT_UAF_F Parameter, 11-8
AUDIT_UAF_P Parameter, 10-15
Automated Fix Parameters
  AUDIT_BREAKIN_F, 11-8
  AUDIT_BYPASS_F, 11-8
  AUDIT_INSTALL_F, 11-8
  AUDIT_LOGFAIL_F, 11-8
  AUDIT_READALL_F, 11-8
  AUDIT_UAF_F, 11-8
  BUGCHECK_F, 11-4
  CAPACITY_F, 11-4
  COMMAND#_F, 11-4
  CPU_F, 11-5
  DEADLOCK_CF, 11-5
  DEADLOCK_F, 11-5
  DISK_CF, 11-5
  DISK_F, 11-5
  ERRFMT_BUGCHECK_F, 11-7
  ERRFMT_BUS_F, 11-7
  ERRFMT_CARD_F, 11-7
  ERRFMT_DISK_F, 11-7
  ERRFMT_HARD_MEMORY_F, 11-7
  ERRFMT_INFORMATION_F, 11-7
  ERRFMT_LINE_PRINTER_F, 11-7
  ERRFMT_MAILBOX_F, 11-7
  ERRFMT_MEMORY_F, 11-7
  ERRFMT_MISCELLANEOUS_F, 11-7
  ERRFMT_POWER_F, 11-7
  ERRFMT_REALTIME_F, 11-7
  ERRFMT_SYNCHRONOUS_COMMUNICATION_F, 11-7
  ERRFMT_SYSTEM_F, 11-7
  ERRFMT_TAPE_F, 11-7
  ERRFMT_TERMINAL_F, 11-7
  ERRFMT_VOLUME_F, 11-7
  ERRFMT_WORKSTATION_F, 11-7
  GBLSECTIONS_CF, 11-5
  GBLSECTIONS_F, 11-5
  IO_CF, 11-5
  IO_F, 11-5
  LOGICAL_NAMES_F, 11-5
  NETWORK#_F, 11-6
  NETWORK_ASCII_F, 11-5
  NETWORK_TIME#_F, 11-6
  PAGE_FAULTS_F, 11-6
  PAGEFILE_CF, 11-6
  PAGEFILE_F, 11-6
  POOL_CF, 11-4, 11-6
  POOL_F, 11-4, 11-6
  POWER_F, 11-6
  XQP_CACHE_F, 11-7
Automated Fix Template Files, 11-3
Automatic Failover, 5-3
AUTOSTART Parameter, 5-3, 5-4, 5-17, 5-25
Binary Event Files
  Reporting, 9-2
  Sending, 9-1
BINARY Parameter, 9-7, 9-16, 14-1
BUGCHECK Parameter, 6-8
BUGCHECK_F Parameter, 11-4
BUGCHECK_P Parameter, 10-11
BURST_DELAY Parameter, 5-25
CAPACITY Parameter, 3-9, 5-29, 6-8
CAPACITY_F Parameter, 11-4

CAPACITY_P Parameter, 10-11, 10-13
CATEGORY Parameter, 9-8
CLEAR PAGER Command, 10-4, 10-6, 10-7, 10-8
CLEAR PAGER Command, 10-10
CLEAR SYSUAF Command, 12-12
Command Interface, 2-3
COMMAND# Parameter, 7-1, 7-2, 7-3, 7-8
COMMAND#_F Parameter, 7-8, 11-4
COMMAND#_F Parameter, 7-3
COMMAND#_P Parameter, 7-3, 7-8, 10-11
Commands
  CLEAR PAGER, 10-4, 10-6, 10-7, 10-8
  CLEAR PAGER, 10-10
  CLEAR SYSUAF, 12-12
  CONNECT, 7-6
  DELETE PAGER, 10-7
  DEMONSTRATE ALIGNMENT_FAULTS, 13-2
  DEMONSTRATE AUDIT_SERVER, 13-3
  DEMONSTRATE AUDIT_SERVER, 12-3
  DEMONSTRATE BREAKIN, 13-4
  DEMONSTRATE BUGCHECK, 13-5
  DEMONSTRATE CPU, 13-6
  DEMONSTRATE DEAD_LOCKS, 13-7
  DEMONSTRATE DISK, 13-8
  DEMONSTRATE ERRORLOG, 13-9
  DEMONSTRATE GLOBAL_SECTIONS, 13-10
  DEMONSTRATE IO, 13-11
  DEMONSTRATE LOGICAL_NAMES, 13-12
  DEMONSTRATE NETWORK_OBJECT, 13-13
  DEMONSTRATE PAGE_FAULTS, 13-14
  DEMONSTRATE PAGR, 13-15
  DEMONSTRATE XQP_CACHE, 13-16
  DISABLE AUTOSTART, 5-3, 5-4, 5-17, 5-25
  EDIT PARAMETERS, 3-8
  ENABLE AUTOSTART, 5-3, 5-4, 5-18
  EXIT, 2-3
  EXIT, 3-5
  HELP, 2-4, B-3
  READ PARAMETERS, 3-6
  SEND ALL, 3-7, 9-1
  SEND CONTACT_INFORMATION, 3-6, 4-1, 4-2, 4-3, 4-4, 9-1
  SEND REQUEST, 3-8, 3-10, 9-1, B-7
  SEND TUNING_INFORMATION, 3-9, 9-1
  SET EDITOR, 3-4, 3-10, 5-4
  SET EDITOR, 5-4
  SET PASSWORD, 8-5
  SET TIME, 8-6
  SHOW EDITOR, 3-10, 5-4
  SHOW EDITOR, 3-11
  SHOW PAGER, 10-4, 10-8
  SHOW PARAMETERS, 3-12
  SHOW STATUS, 12-12
  SHOW STATUS, 5-19
  SHOW TEMPERATURE, 6-3, 6-25
  SHOW TIME, 8-7
  START MONITOR, 3-6, 5-20, 7-3
  START TRANSPORT, 3-6, 5-21
  STOP MONITOR, 3-6, 5-22, 7-3
  STOP TRANSPORT, 5-23
  SUMMARIZE, 14-1, 14-4, 14-6
  WAKE TRANSPORT, 5-24
  WAKE TRANSPORT, 5-2
Comments, 3-1
CONFIGURE Parameter, 6-10
CONNECT Command, 7-6
Contact Name Parameter, 4-3
Control File
  OSM_CTLFILE.DAT, 3-1
  OSM_CTLFILE_nodename.DAT, 3-1
CPU Parameter, 6-10
CPU_F Parameter, 11-5
CPU_P Parameter, 10-12
CPU_WAKE Parameter, 5-25, 6-1, 6-16, 6-19, B-7
CRITICAL Parameter, 5-26, 6-1, 6-4, 7-2, 9-1, 9-9
Data Reduction, 9-2
DEADLOCK Parameter, 6-12
DEADLOCK_C Parameter, 6-12, 10-11
DEADLOCK_CF Parameter, 11-5
DEADLOCK_F Parameter, 11-5
DEADLOCK_P Parameter, 10-11
DEADLOCK_P Parameter, 10-12
DECNET_OBJECT Parameter, 8-8
DECNET_OBJECT=300,OSM_MON Parameter, 12-18
DEFAULT_NETWORK Parameter, 8-8
DELETE PAGER Command, 10-7
DEMONSTRATE ALIGNMENT_FAULTS, 13-2
DEMONSTRATE AUDIT_SERVER Command, 12-3, 13-3
DEMONSTRATE BREAKIN Command, 13-4
DEMONSTRATE BUGCHECK, 13-5
DEMONSTRATE CPU, 13-6
DEMONSTRATE DEAD_LOCKS, 13-7
DEMONSTRATE DISK, 13-8
DEMONSTRATE ERRORLOG, 13-9
DEMONSTRATE GLOBAL_SECTIONS, 13-10
DEMONSTRATE IO, 13-11
DEMONSTRATE LOGICAL_NAMES, 13-12
DEMONSTRATE NETWORK_OBJECT, 13-13
DEMONSTRATE PAGE_FAULTS, 13-14
DEMONSTRATE PAGR, 13-15
DEMONSTRATE XQP_CACHE, 13-16
DETAIL Parameter, 4-1, 4-3, 4-4, 9-3, 9-9, 10-9, 11-3
DETAIL Parameter, 9-16, 9-17
DETAIL_ERRORLOG Parameter, 9-11
DETAIL_ERRORLOG Parameter, 9-16, 9-17
DIAGNOSE Parameter, 9-11, 9-16, 9-17
DISABLE AUTOSTART Command, 5-3, 5-4, 5-17, 5-25
DISABLED Parameter, 9-11
DISK Parameter, 5-6, 6-13
DISK_C Parameter, 5-6, 6-13
DISK_CF Parameter, 5-6, 11-5
DISK_F Parameter, 5-6, 11-5
DISK_P Parameter, 5-6
DISK_P Parameter, 10-12
EDIT PARAMETERS Command, 3-8
EDITOR Parameter, 3-4, 3-10, 5-4, 5-26
ELV Parameter, 9-16, 9-17
ENABLE AUTOSTART Command, 5-3, 5-4, 5-18
ENABLED Parameter, 9-12
ENHANCE Parameter, 14-7
ERRFMT_BUGCHECK Parameter, 6-14
ERRFMT_BUGCHECK_F Parameter, 11-7
ERRFMT_BUGCHECK_P Parameter, 10-14
ERRFMT_BUS Parameter, 6-14
ERRFMT_BUS_F Parameter, 11-7
ERRFMT_BUS_P Parameter, 10-14
ERRFMT_CARD Parameter, 6-14

ERRFMT_CARD_F Parameter, 11-7
ERRFMT_CARD_P Parameter, 10-14
ERRFMT_DISK Parameter, 6-14
ERRFMT_DISK_F Parameter, 11-7
ERRFMT_DISK_P Parameter, 10-14
ERRFMT_HARD_MEMORY Parameter, 6-14
ERRFMT_HARD_MEMORY_F Parameter, 11-7
ERRFMT_HARD_MEMORY_P Parameter, 10-14
ERRFMT_INFORMATION Parameter, 6-14
ERRFMT_INFORMATION_F Parameter, 11-7
ERRFMT_INFORMATION_P Parameter, 10-14
ERRFMT_LINE_PRINTER Parameter, 6-14
ERRFMT_LINE_PRINTER_F Parameter, 11-7
ERRFMT_LINE_PRINTER_P Parameter, 10-14
ERRFMT_MAILBOX Parameter, 6-14
ERRFMT_MAILBOX_F Parameter, 11-7
ERRFMT_MAILBOX_P Parameter, 10-14
ERRFMT_MEMORY Parameter, 6-14
ERRFMT_MEMORY_F Parameter, 11-7
ERRFMT_MEMORY_P Parameter, 10-14
ERRFMT_MISCELLANEOUS Parameter, 6-14
ERRFMT_MISCELLANEOUS_F Parameter, 11-7
ERRFMT_MISCELLANEOUS_P Parameter, 10-14
ERRFMT_POWER Parameter, 6-14
ERRFMT_POWER_F Parameter, 11-7
ERRFMT_POWER_P Parameter, 10-14
ERRFMT_REALTIME Parameter, 6-14
ERRFMT_REALTIME_F Parameter, 11-7
ERRFMT_REALTIME_P Parameter, 10-14
ERRFMT_SYNCHRONOUS_COMMUNICATION Parameter, 6-15
ERRFMT_SYNCHRONOUS_COMMUNICATION_F Parameter, 11-7
ERRFMT_SYNCHRONOUS_COMMUNICATION_P Parameter, 10-14
ERRFMT_SYSTEM Parameter, 6-15
ERRFMT_SYSTEM_F Parameter, 11-7
ERRFMT_SYSTEM_P Parameter, 10-14
ERRFMT_TAPE Parameter, 6-15
ERRFMT_TAPE_F Parameter, 11-7
ERRFMT_TAPE_P Parameter, 10-14
ERRFMT_TERMINAL Parameter, 6-15
ERRFMT_TERMINAL_F Parameter, 11-7
ERRFMT_TERMINAL_P Parameter, 10-14
ERRFMT_VOLUME Parameter, 6-15
ERRFMT_VOLUME_F Parameter, 11-7
ERRFMT_VOLUME_P Parameter, 10-14
ERRFMT_WORKSTATION Parameter, 6-15
ERRFMT_WORKSTATION_F Parameter, 11-7
ERRFMT_WORKSTATION_P Parameter, 10-14
ERRORLOG Parameter, 9-13
Errorlog Thresholds, 6-2
ERRORLOG_FORMAT Parameter, 6-14, 9-13, 9-16, 9-17
ERRORLOG_TRANSLATE Parameter, 6-14, 9-13
Events
  Monitored, 1-7
Exception Event Parameters
  AGE_ACCOUNTING, 6-5
  AGE_ERRORLOG, 6-5
  AGE_OPERATOR, 6-5
  ALIGNMENT_FAULTS, 6-6
  AUDIT_BREAKIN, 6-7
  AUDIT_BYPASS, 6-7
  AUDIT_INSTALL, 6-7
  AUDIT_LOGFAIL, 6-7
  AUDIT_READALL, 6-7
  AUDIT_UAF, 6-7
  BUGCHECK, 6-8
  BUGCHECK_P, 10-11
  CAPACITY, 3-9, 5-29, 6-8
  COMMAND#, 7-1, 7-2, 7-3, 7-8
  COMMAND#_F, 7-3, 7-8
  COMMAND#_P, 7-3, 7-8
  CPU, 6-10
  CRITICAL, 5-26, 6-1, 6-4, 7-2, 9-1, 9-9
  DEADLOCK, 6-12
  DEADLOCK_C, 6-12, 10-11
  DEADLOCK_P, 10-11
  DISK, 5-6, 6-13
  DISK_C, 5-6, 6-13
  DISK_CF, 5-6
  DISK_F, 5-6
  DISK_P, 5-6
  ERRFMT_BUGCHECK, 6-14
  ERRFMT_BUS, 6-14
  ERRFMT_CARD, 6-14
  ERRFMT_DISK, 6-14
  ERRFMT_HARD_MEMORY, 6-14
  ERRFMT_INFORMATION, 6-14
  ERRFMT_LINE_PRINTER, 6-14
  ERRFMT_MAILBOX, 6-14
  ERRFMT_MEMORY, 6-14
  ERRFMT_MISCELLANEOUS, 6-14
  ERRFMT_POWER, 6-14
  ERRFMT_REALTIME, 6-14
  ERRFMT_SYNCHRONOUS_COMMUNICATION, 6-15
  ERRFMT_SYSTEM, 6-15
  ERRFMT_TAPE, 6-15
  ERRFMT_TERMINAL, 6-15
  ERRFMT_VOLUME, 6-15
  ERRFMT_WORKSTATION, 6-15
  GBLSECTIONS, 6-15
  GBLSECTIONS_C, 6-15
  IO, 6-16
  IO_C, 6-16
  LOGICAL_NAMES, 6-18
  NETWORK#, 8-9
  NETWORK_ASCII, 8-10
  NETWORK_TIME#, 8-9
  NETWORK_TIME_SET#, 8-10
  PAGE_FAULTS, 6-19
  PAGEFILE, 6-21
  PAGEFILE_C, 6-21
  POOL, 6-22
  POOL_C, 6-9, 6-22, 6-24
  POWER, 6-23
  POWER_P, 10-3
  TEMPERATURE, 6-25
  TEMPERATURE_C, 6-25
  TIME_SET, 8-11

XQP_CACHE, 6-26
EXIT Command, 2-3
EXIT Command, 3-5
EXIT Parameter, 3-13
Failover
Transport Process, 1-5, 5-3
FAX Parameter, 4-3
FILE Parameter, 9-14, 9-16
GBLSECTIONS Parameter, 6-15
GBLSECTIONS_C Parameter, 6-15
GBLSECTIONS_CF Parameter, 11-5
GBLSECTIONS_F Parameter, 11-5
GBLSECTIONS_P Parameter, 10-12
Hardware Requirements, 1-12
HBOUTPUT Parameter, 9-14, 9-16
HEARTBEAT Parameter, 5-26, 9-9
HELP Command, 2-4, B-3
Identification Parameters
ACCOUNT, 4-3, 9-13
Address, 4-3
Contact Name, 4-3
FAX, 4-3
Organization Name, 4-3
PHONE, 4-4
SITE, 4-4, B-6, B-7
IO Parameter, 6-16
IO_C Parameter, 6-16
IO_CF Parameter, 11-5
IO_F Parameter, 11-5
IO_P Parameter, 10-12
LOG Parameter, 5-26, 11-3
Logical Names, 5-4
OSM$, 5-4
OSM$EDIT, 3-4, 3-10, 3-11, 5-4
OSM$LOCAL, 5-5, 5-7, 7-1, 7-2, 7-3, 7-4, B-1
OSM$LOG, 5-5
OSM_*, 5-5
OSM_AUDIT_LISTENER, 5-5
OSM_DISK_ *, 5-6
OSM_ERRFMT, 5-7
OSM_LOGICALS.TEMPLATE file, 5-7
OSM_SYSUAF_CLASS, 12-9
OSM_SYSUAF_CLASS_nodename, 12-9
OSM_SYSUAF_FIELD_class_n, 12-10
OSM_SYSUAF_USER_class_n, 12-11
OSM_VER, 5-7
OSM_VMS_VER, 5-7
Template, 5-7
LOGICAL_NAMES Parameter, 6-18
LOGICAL_NAMES_F Parameter, 11-5
LOGICAL_NAMES_P Parameter, 10-12
Logicals
OSM_SYSUAF_CLASS, 12-9
OSM_SYSUAF_FIELD_CLASS_N, 12-3, 12-11
OSM_SYSUAF_USER_CLASS_N, 12-3
MAIL Parameter, 9-14, 9-17
MAIL Parameter, 9-17
MONITOR_WAKE Parameter, 5-28, 6-1, 7-1
Monitored Events Summary, 1-7
Network Event Monitoring
Commands, 8-5
Network Event Monitoring Parameters, 8-8
NETWORK# Parameter, 8-9
NETWORK#_F Parameter, 11-6
NETWORK#_P Parameter, 10-12
NETWORK_ASCII Parameter, 8-10
NETWORK_ASCII_F Parameter, 11-5
NETWORK_ASCII_P Parameter, 10-12
NETWORK_ID Parameter, 8-10
NETWORK_TIME# Parameter, 8-9
NETWORK_TIME#_F Parameter, 11-6
NETWORK_TIME#_P Parameter, 10-13
NETWORK_TIME_SET# Parameter, 8-10
NIGHT Parameter, 5-28, 9-1
NORMAL Parameter, 14-7
OPCOM Parameter, 5-28
OPCOM_ERROR Parameter, 5-28
OPCOM_ERROR_P Parameter, 10-13
OPERATOR Parameter, 9-15
OPERATOR Parameter, 9-17
Organization Name Parameter, 4-3
OSM_SYSUAF_CLASS Logical, 12-9
OSM_SYSUAF_FIELD_CLASS_N Logical, 12-3, 12-11
OSM_SYSUAF_USER_CLASS_N Logical, 12-3
OUTPUT Parameter, 5-29, 9-2, 9-7, 9-9, 9-11, 9-13, 9-14, 9-16, 9-17, 9-19, 9-20, 9-21, 10-1, 10-2, 10-9, 14-1
OUTPUT_ACCOUNTS Parameter, 9-17
OUTPUT_REASONS Parameter, 9-18
OUTPUT_SEVERITY Parameter, 9-19
PAGE_FAULTS Parameter, 6-19
PAGE_FAULTS_F Parameter, 11-6
PAGE_FAULTS_P Parameter, 10-13
PAGEFILE Parameter, 6-21
PAGEFILE_C Parameter, 6-21
PAGEFILE_CF Parameter, 11-6
PAGEFILE_F Parameter, 11-6
PAGEFILE_P Parameter, 10-13
PAGER Parameter, 10-1, 10-2, 10-10
Pager Parameters
AUDIT_BREAKIN_P, 10-15
AUDIT_BYPASS_P, 10-15
AUDIT_INSTALL_P, 10-15
AUDIT_LOGFAIL_P, 10-15
AUDIT_READALL_P, 10-15
AUDIT_UAF_P, 10-15
BUGCHECK_P, 10-11
CAPACITY_P, 10-11, 10-13
COMMAND#_P, 10-11
CPU_P, 10-12
DEADLOCK_P, 10-12
DISK_P, 10-12
ERRFMT_BUS_P, 10-14
ERRFMT_CARD_P, 10-14
ERRFMT_DISK_P, 10-14
ERRFMT_HARD_MEMORY_P, 10-14
ERRFMT_INFORMATION_P, 10-14
ERRFMT_LINE_PRINTER_P, 10-14
ERRFMT_MAILBOX_P, 10-14
ERRFMT_MEMORY_P, 10-14
ERRFMT_MISCELLANEOUS_P, 10-14
ERRFMT_POWER_P, 10-14
ERRFMT_REALTIME_P, 10-14
ERRFMT_SYNCHRONOUS_COMMUNICATION_P, 10-14
ERRFMT_SYSTEM_P, 10-14
ERRFMT_TAPE_P, 10-14

ERRFMT_TERMINAL_P, 10-14
ERRFMT_VOLUME_P, 10-14
ERRFMT_WORKSTATION_P, 10-14
GBLSECTIONS_P, 10-12
IO_P, 10-12
LOGICAL_NAMES_P, 10-12
NETWORK#_P, 10-12
NETWORK_ASCII_P, 10-12
NETWORK_TIME#_P, 10-13
OPCOM_ERROR_P, 10-13
PAGE_FAULTS_P, 10-13
PAGEFILE_P, 10-13
Pager, 10-1, 10-2, 10-10
PAGER_DELETE, 5-28, 10-10, B-4
PAGER_RETRY, 10-4, 10-5, 10-7, 10-8
PAGER_RETRY, 10-10
POOL_P, 10-13
POWER_P, 10-13
XQP_CACHE_P, 10-13
PAGER_DELETE Parameter, 5-28, 10-10, B-4
PAGER_RETRY Parameter, 10-4, 10-5, 10-7, 10-8
PAGER_RETRY Parameter, 10-10
Paging
Backup Paging, 10-5
Clearing a page, 10-4
Generation of a Page, 10-2
Notification String, 10-3
Paging History, 10-5
Repeating a page, 10-4
Paging Parameters
ERRFMT_BUGCHECK_P, 10-14
Parameter Formats, 3-2
Parameters
ACCOUNT, 4-3, 9-13
ACTIVE, 9-6
Address, 4-3
AGE_ACCOUNTING, 6-5
AGE_ERRORLOG, 6-5
AGE_OPERATOR, 6-5
ALIGNMENT_FAULTS, 6-6
ANALYZE/ERROR, 9-6, 9-16, 9-17
ANALYZE/ERROR, 9-17
ANALYZE/ERROR/ELV, 9-12
AUDIT, 6-7
AUDIT, 12-18
Audit Server Automated Fix, 11-8
Audit Server Message Type, 6-7
Audit Server Paging, 10-15
AUDIT_BREAKIN, 6-7
AUDIT_BREAKIN_F, 11-8
AUDIT_BREAKIN_P, 10-15
AUDIT_BYPASS, 6-7
AUDIT_BYPASS_F, 11-8
AUDIT_BYPASS_P, 10-15
AUDIT_INSTALL, 6-7
AUDIT_INSTALL_F, 11-8
AUDIT_INSTALL_P, 10-15
AUDIT_LOGFAIL, 6-7
AUDIT_LOGFAIL_F, 11-8
AUDIT_LOGFAIL_P, 10-15
AUDIT_READALL, 6-7
AUDIT_READALL_F, 11-8
AUDIT_READALL_P, 10-15
AUDIT_UAF, 6-7
AUDIT_UAF_F, 11-8
AUDIT_UAF_P, 10-15
AUTOSTART, 5-3, 5-4, 5-17, 5-25
AUTOSTART, 5-3, 5-18
BINARY, 9-7, 9-16, 14-1
BUGCHECK, 6-8
BUGCHECK_F, 11-4
BUGCHECK_P, 10-11
BURST_DELAY, 5-25
CAPACITY, 3-9, 5-29, 6-8
CAPACITY_F, 11-4
CAPACITY_P, 10-11, 10-13
CATEGORY, 9-8
COMMAND#, 7-1, 7-2, 7-3, 7-8
COMMAND#_F, 7-3, 7-8, 11-4
COMMAND#_P, 7-3, 7-8, 10-11
CONFIGURE, 6-10
Contact Name, 4-3
CPU, 6-10
CPU_F, 11-5
CPU_P, 10-12
CPU_WAKE, 5-25, 6-16, 6-19, B-7
CPU_WAKE, 6-1
CRITICAL, 5-26, 6-1, 6-4, 7-2, 9-1, 9-9
DEADLOCK, 6-12
DEADLOCK_C, 6-12, 10-11
DEADLOCK_CF, 11-5
DEADLOCK_F, 11-5
DEADLOCK_P, 10-11
DEADLOCK_P, 10-12
DECNET_OBJECT, 8-8
DECNET_OBJECT=300,OSM_MON, 12-18
DEFAULT_NETWORK, 8-8
DETAIL, 4-1, 4-3, 4-4, 9-3, 9-9, 10-9, 11-3
DETAIL, 9-16, 9-17
DETAIL_ERRORLOG, 9-11
DETAIL_ERRORLOG, 9-16, 9-17
DIAGNOSE, 9-11, 9-16, 9-17
DISABLED, 9-11
DISK, 5-6, 6-13
DISK_C, 5-6, 6-13
DISK_CF, 5-6, 11-5
DISK_F, 5-6, 11-5
DISK_P, 5-6
DISK_P, 10-12
EDITOR, 3-4, 3-10, 5-4, 5-26
ELV, 9-16, 9-17
ENABLED, 9-12
ENHANCE, 14-7
ERRFMT, 6-14
ERRFMT_BUGCHECK, 6-14
ERRFMT_BUGCHECK_F, 11-7
ERRFMT_BUGCHECK_P, 10-14
ERRFMT_BUS, 6-14
ERRFMT_BUS_F, 11-7
ERRFMT_BUS_P, 10-14
ERRFMT_CARD, 6-14
ERRFMT_CARD_F, 11-7
ERRFMT_CARD_P, 10-14
ERRFMT_DISK, 6-14

ERRFMT_DISK_F, 11-7
ERRFMT_DISK_P, 10-14
ERRFMT_HARD_MEMORY, 6-14
ERRFMT_HARD_MEMORY_F, 11-7
ERRFMT_HARD_MEMORY_P, 10-14
ERRFMT_INFORMATION, 6-14
ERRFMT_INFORMATION_F, 11-7
ERRFMT_INFORMATION_P, 10-14
ERRFMT_LINE_PRINTER, 6-14
ERRFMT_LINE_PRINTER_F, 11-7
ERRFMT_LINE_PRINTER_P, 10-14
ERRFMT_MAILBOX, 6-14
ERRFMT_MAILBOX_F, 11-7
ERRFMT_MAILBOX_P, 10-14
ERRFMT_MEMORY, 6-14
ERRFMT_MEMORY_F, 11-7
ERRFMT_MEMORY_P, 10-14
ERRFMT_MISCELLANEOUS, 6-14
ERRFMT_MISCELLANEOUS_F, 11-7
ERRFMT_MISCELLANEOUS_P, 10-14
ERRFMT_POWER, 6-14
ERRFMT_POWER_F, 11-7
ERRFMT_POWER_P, 10-14
ERRFMT_REALTIME, 6-14
ERRFMT_REALTIME_F, 11-7
ERRFMT_REALTIME_P, 10-14
ERRFMT_SYNCHRONOUS_COMMUNICATION, 6-15
ERRFMT_SYNCHRONOUS_COMMUNICATION_F, 11-7
ERRFMT_SYNCHRONOUS_COMMUNICATION_P, 10-14
ERRFMT_SYSTEM, 6-15
ERRFMT_SYSTEM_F, 11-7
ERRFMT_SYSTEM_P, 10-14
ERRFMT_TAPE, 6-15
ERRFMT_TAPE_F, 11-7
ERRFMT_TAPE_P, 10-14
ERRFMT_TERMINAL, 6-15
ERRFMT_TERMINAL_F, 11-7
ERRFMT_TERMINAL_P, 10-14
ERRFMT_VOLUME, 6-15
ERRFMT_VOLUME_F, 11-7
ERRFMT_VOLUME_P, 10-14
ERRFMT_WORKSTATION, 6-15
ERRFMT_WORKSTATION_F, 11-7
ERRFMT_WORKSTATION_P, 10-14
Error Format Automated Fix, 11-7
Error Format Message Types, 6-14
Error Format Paging, 10-14
ERRORLOG, 9-13
ERRORLOG_FORMAT, 6-14, 9-13, 9-16, 9-17
ERRORLOG_TRANSLATE, 6-14, 9-13
EXIT, 3-13
FAX, 4-3
FILE, 9-14, 9-16
GBLSECTIONS, 6-15
GBLSECTIONS_C, 6-15
GBLSECTIONS_CF, 11-5
GBLSECTIONS_F, 11-5
GBLSECTIONS_P, 10-12
HBOUTPUT, 9-14, 9-16
HEARTBEAT, 5-26, 9-9
IO, 6-16
IO_C, 6-16
IO_CF, 11-5
IO_F, 11-5
IO_P, 10-12
LOG, 5-26, 11-3
LOGICAL_NAMES, 6-18
LOGICAL_NAMES_F, 11-5
LOGICAL_NAMES_P, 10-12
MAIL, 9-14, 9-17
MAIL, 9-17
MONITOR_WAKE, 5-28, 6-1, 7-1
NETWORK#, 8-9
NETWORK#_F, 11-6
NETWORK#_P, 10-12
NETWORK_ASCII, 8-10
NETWORK_ASCII_F, 11-5
NETWORK_ASCII_P, 10-12
NETWORK_ID, 8-10
NETWORK_TIME#, 8-9
NETWORK_TIME#_F, 11-6
NETWORK_TIME#_P, 10-13
NETWORK_TIME_SET#, 8-10
NIGHT, 5-28, 9-1
NORMAL, 14-7
OPCOM, 5-28
OPCOM_ERROR, 5-28
OPCOM_ERROR_P, 10-13
OPERATOR, 9-15
OPERATOR, 9-17
Organization Name, 4-3
OUTPUT, 5-29, 9-2, 9-7, 9-9, 9-11, 9-13, 9-14, 9-17, 9-19, 9-20, 9-21, 10-1, 10-2, 10-9, 14-1
OUTPUT, 9-16
OUTPUT_ACCOUNTS, 9-17
OUTPUT_REASONS, 9-18
OUTPUT_SEVERITY, 9-19
PAGE_FAULTS, 6-19
PAGE_FAULTS_F, 11-6
PAGE_FAULTS_P, 10-13
PAGEFILE, 6-21
PAGEFILE_C, 6-21
PAGEFILE_CF, 11-6
PAGEFILE_F, 11-6
PAGEFILE_P, 10-13
PAGER, 10-1, 10-2, 10-10
PAGER_DELETE, 5-28, 10-10, B-4
PAGER_RETRY, 10-4, 10-5, 10-7, 10-8
PAGER_RETRY, 10-10
PHONE, 4-4
POOL, 6-22
POOL_C, 6-9, 6-22, 6-24
POOL_CF, 11-4, 11-6
POOL_F, 11-4, 11-6
POOL_P, 10-13
POWER, 6-23
POWER_F, 11-6
POWER_P, 10-3
POWER_P, 10-13
PRIME, 5-29, 14-9, 14-10, 14-11
PRINTER, 9-17, 9-19

PURGE_WORKING_SET, 5-2, 5-29
REPEAT_ALWAYS, 9-2, 9-19
REPEAT_DOM, 5-28, 9-2, 9-19
REPEAT_DOW, 9-2, 9-19
REPEAT_DOW, 5-28
SCRIPT, 9-17, 9-20
SITE, 4-4, B-6, B-7
SYSUAF, 12-2, 12-17
SYSUAF_ALLOW, 12-2, 12-17
SYSUAF_DISTRIBUTE, 12-18
SYSUAF_DISTRIBUTE_ADD, 12-18
SYSUAF_DISTRIBUTE_VERIFY, 12-18
SYSUAF_DISTRIBUTE_VERIFY_ADD, 12-18
SYSUAF_NODE, 12-6, 12-18
SYSUAF_NOREMOVE, 12-18
SYSUAF_NOTIFY, 12-18
TCPIP_PORT, 8-11
TEMPERATURE, 6-25
TEMPERATURE_C, 6-25
TEXT_FORMAT, 9-20
TEXT_MODE, 9-21
TEXT_POSTFIX, 9-21
TEXT_POSTFIX_ADD, 9-21
TEXT_PREFIX, 9-21
TEXT_PREFIX_ADD, 9-21
TIME_SET, 8-11
TRANSPORT_WAKE, 5-2, 5-29, 9-1
TRANSPORT_WAKE, 10-10
UIC, 5-30
WSEA, 9-16
XQP_CACHE, 6-26
XQP_CACHE_F, 11-7
XQP_CACHE_P, 10-13
PHONE Parameter, 4-4
POOL Parameter, 6-22
POOL_C Parameter, 6-9, 6-22, 6-24
POOL_CF Parameter, 11-4, 11-6
POOL_F Parameter, 11-4, 11-6
POOL_P Parameter, 10-13
POWER Parameter, 6-23
POWER_F Parameter, 11-6
POWER_P Parameter, 10-3
POWER_P Parameter, 10-13
PRIME Parameter, 14-9, 14-10, 14-11
PRIME Parameter, 5-29
PRINTER Parameter, 9-17, 9-19
Privileges, 1-13
PURGE_WORKING_SET Parameter, 5-2, 5-29
PURGE_WORKING_SET Parameter, 5-2
Purging Working Sets, 5-2
READ PARAMETERS Command, 3-6
REPEAT_ALWAYS Parameter, 9-2, 9-19
REPEAT_DOM Parameter, 5-28, 9-2, 9-19
REPEAT_DOW Parameter, 9-2, 9-19
REPEAT_DOW Parameter, 5-28
Reporting on Binary Event Files, 9-2
Reporting Parameters
ACTIVE, 9-6
ANALYZE/ERROR, 9-6, 9-16, 9-17
ANALYZE/ERROR, 9-17
ANALYZE/ERROR/ELV, 9-12
BINARY, 9-7, 9-16, 14-1
CATEGORY, 9-8
DETAIL, 4-1, 4-3, 4-4, 9-3, 9-9, 10-9, 11-3
DETAIL, 9-16, 9-17
DETAIL_ERRORLOG, 9-11
DETAIL_ERRORLOG, 9-16, 9-17
DIAGNOSE, 9-11, 9-16, 9-17
DISABLED, 9-11
ELV, 9-16, 9-17
ENABLED, 9-12
ENHANCE, 14-7
ERRORLOG, 9-13
ERRORLOG_FORMAT, 6-14, 9-13, 9-16, 9-17
ERRORLOG_TRANSLATE, 6-14, 9-13
FILE, 9-14, 9-16
HBOUTPUT, 9-14, 9-16
MAIL, 9-14, 9-17
MAIL, 9-17
NORMAL, 14-7
OPERATOR, 9-15, 9-17
OUTPUT, 5-29, 9-2, 9-7, 9-9, 9-11, 9-13, 9-14, 9-16, 9-17, 9-19, 9-20, 9-21, 10-1, 10-2, 10-9, 14-1
OUTPUT_ACCOUNTS, 9-17
OUTPUT_REASONS, 9-18
OUTPUT_SEVERITY, 9-19
PRINTER, 9-17, 9-19
REPEAT_ALWAYS, 9-2, 9-19
REPEAT_DOM, 5-28, 9-2, 9-19
REPEAT_DOW, 9-2, 9-19
REPEAT_DOW, 5-28
SCRIPT, 9-17, 9-20
TEXT_FORMAT, 9-20
TEXT_MODE, 9-21
TEXT_POSTFIX, 9-21
TEXT_POSTFIX_ADD, 9-21
TEXT_PREFIX, 9-21
TEXT_PREFIX_ADD, 9-21
WSEA, 9-16
SCRIPT Parameter, 9-17, 9-20
Security alarms, 5-2, 5-28, 6-1, 6-7, 9-3, 10-15, 11-8, 12-2, 13-3
SEND ALL Command, 3-7, 9-1
SEND CONTACT_INFORMATION Command, 3-6, 4-1, 4-2, 4-3, 4-4, 9-1
SEND REQUEST Command, 3-8, 3-10, 9-1, B-7
SEND TUNING_INFORMATION Command, 3-9, 9-1
Sending Binary Event Files, 9-1
SET EDITOR Command, 3-4, 3-10, 5-4
SET PASSWORD Command, 8-5
SET TIME Command, 8-6
Severity levels, 6-4
SHOW EDITOR Command, 3-10, 5-4
SHOW EDITOR Command, 3-11
SHOW PAGER Command, 10-4, 10-8
SHOW PARAMETERS Command, 3-12
SHOW STATUS Command, 12-12
SHOW STATUS Command, 5-19
SHOW TEMPERATURE Command, 6-3, 6-25
SHOW TIME Command, 8-7
SITE Parameter, 4-4, B-6, B-7
Software Requirements, 1-12
START MONITOR Command, 3-6, 5-20, 7-3
START TRANSPORT Command, 3-6, 5-21
Starting OSM, 5-1
STOP MONITOR Command, 3-6, 5-22, 7-3
STOP TRANSPORT Command, 5-23
Stopping OSM, 5-1
SUMMARIZE Command, 14-1, 14-4, 14-6

Summary
OSM_SUMMARY_SETUP.COM, 14-1
OSM_SUMMARY_UDM.TEMPLATE, 14-2
SYSUAF Parameter, 12-2, 12-17
SYSUAF Synchronization, 12-1
SYSUAF_ALLOW Parameter, 12-2, 12-17
SYSUAF_DISTRIBUTE Parameter, 12-18
SYSUAF_DISTRIBUTE_ADD Parameter, 12-18
SYSUAF_DISTRIBUTE_VERIFY Parameter, 12-18
SYSUAF_DISTRIBUTE_VERIFY_ADD Parameter, 12-18
SYSUAF_NODE Parameter, 12-6
SYSUAF_NODE Parameter, 12-18
SYSUAF_NOREMOVE Parameter, 12-18
SYSUAF_NOTIFY Parameter, 12-18
TCPIP_PORT Parameter, 8-11
TEMPERATURE Parameter, 6-25
TEMPERATURE_C Parameter, 6-25
Template
Automated Fix, 11-3
OSM Logical Names, 5-7
User Defined Event Monitoring, 7-4
TEXT_FORMAT Parameter, 9-20
TEXT_MODE Parameter, 9-21
TEXT_POSTFIX Parameter, 9-21
TEXT_POSTFIX_ADD Parameter, 9-21
TEXT_PREFIX Parameter, 9-21
TEXT_PREFIX_ADD Parameter, 9-21
Thresholds, 6-2
TIME_SET Parameter, 8-11
Transport
ASCII Detail, 9-3
Automatic Failover, 5-3
Data Reduction, 9-2
Transport Process
Reporting on Binary Event Files, 9-2
Sending Binary Event Files, 9-1
TRANSPORT_WAKE Parameter, 5-2, 5-29, 9-1
TRANSPORT_WAKE Parameter, 10-10
UIC Parameter, 5-30
User Defined Event Monitoring
Example, 7-4
Operation, 7-1, 8-1
Parameters, 7-8
Setting up, 7-2, 8-1
Template, 7-4
WAKE TRANSPORT Command, 5-24
WSEA Parameter, 9-16
XQP_CACHE Parameter, 6-26
XQP_CACHE_F Parameter, 11-7
XQP_CACHE_P Parameter, 10-13