```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

Copyright (c) 2019 CA. All rights reserved.

This documentation (the "Documentation") and related computer software program (the "Software") (hereinafter collectively referred to as the "Product") is for the end user's informational purposes only and is subject to change or withdrawal by CA at any time.

This Product may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Product is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of the Documentation for their own internal use, and may make one copy of the Software as reasonably required for back-up and disaster recovery purposes, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the provisions of the license for the Software are permitted to have access to such copies.

This right to print copies of the Documentation and to make a copy of the Software is limited to the period during which the license for the Product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to certify in writing to CA that all copies and partial copies of the Product have been returned to CA or destroyed.

EXCEPT AS OTHERWISE STATED IN THE APPLICABLE LICENSE AGREEMENT, TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS PRODUCT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS PRODUCT, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

The use of any product referenced in the Documentation is governed by the end user's applicable license agreement.

The manufacturer of this Product is CA.

This Product is provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Section 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7013(c)(1)(ii), as applicable, or their successors.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

GENERAL TABLE OF CONTENTS

	PAGE
CHAPTER 1. OVERVIEW.	1-001
1.1 Primary Areas of Application	1-002
1.2 Reporting and Inquiry Facilities	1-003
1.3 Files Overview	1-003
1.4 Product Prerequisites.	1-005
CHAPTER 2. USAGE GUIDELINES.	2-001
2.1 Data Analysis.	2-001
2.2 CA MICS Product Interfaces	2-002
CHAPTER 3. REPORTS	3-001
3.1 MICF Inquiries for VAX/VMS	3-001
CHAPTER 4. EXCEPTIONS.	4-001
Chapter 5. FILES	5-001
5.1 Data Element Naming Conventions.	5-002
5.2 VAX/VMS Accounting Information Area Files.	5-003
5.3 VAX/VMS Monitor Information Area Files	5-019
5.4 VAX/VMS System Usage Information Area Files.	5-045
Chapter 6. DATA SOURCES.	6-001
6.1 Data Collection and Transport to CA MICS	6-001
6.2 VMS ACCOUNTING Record Descriptions	6-009
6.3 VMS MONITOR Record Descriptions.	6-012
6.4 CA MICS VMS System Usage (DEXSUS) Record Descriptions	6-023
Chapter 7. DEFINING PARAMETERS	7-001
7.1 Environmental Considerations	7-001
7.2 Complex Level Parameters	7-002
7.3 Unit Level Parameters.	7-011
7.4 Other Related Parameters	7-042
CHAPTER 8. INSTALLATION.	8-001
8.1 Installing the DEXSUS.COM Collection Routine	8-001
Chapter 9. PROCESSING.	9-001
9.1 Processing Considerations.	9-001
9.2 Daily Update Processing Flow	9-002
9.3 User Exits	9-005
9.4 Using COMMON Input DDnames and DEXINPUT exits.	9-006
Chapter 10. MODIFICATION.	1-001
10.1 Parameter Modification.	1-001
APPENDIX A. MESSAGES	A-001
APPENDIX B. DATA DICTIONARY.	B-001

GENERAL TABLE OF CONTENTS

	PAGE
B.1 Common Data Elements	B-001
B.2 VAX/VMS Accounting Information (DEA) Area.	B-027
B.3 VAX/VMS Monitor Information (DEM) Area	B-050
B.4 VAX/VMS System Usage Information (DES) Area.	B-210

DETAIL TABLE OF CONTENTS PAGE

CHAPTER 1. OVERVIEW. 1-001

1.1 Primary Areas of Application 1-002

1.2 Reporting and Inquiry Facilities 1-003

1.3 Files Overview 1-003

1.4 Product Prerequisites. 1-005

CHAPTER 2. USAGE GUIDELINES. 2-001

2.1 Data Analysis. 2-001

2.2 CA MICS Product Interfaces 2-002
2.2.1 Accounting and Chargeback Interface. 2-003
2.2.2 Capacity Planner Interface 2-004

CHAPTER 3. REPORTS 3-001

3.1 MICF Inquiries for VAX/VMS 3-001
3.1.1 VAX/VMS Workload Analysis Inquiries. 3-003
3.1.2 VAX/VMS Performance Analysis Inquiries 3-011
3.1.3 VAX/VMS Disk Storage Analysis Inquiries. . . . 3-027

CHAPTER 4. EXCEPTIONS. 4-001

Chapter 5. FILES 5-001

5.1 Data Element Naming Conventions. 5-002

5.2 VAX/VMS Accounting Information Area Files. . . . 5-003
5.2.1 VMS ACCOUNTING Process Termination File (DEAPRC) 5-003
5.2.1.1 DEAPRC File Organization 5-004
5.2.1.2 DEAPRC Data Elements List. 5-004
5.2.1.3 DEAPRC Usage Considerations. 5-005
5.2.1.4 DEAPRC Retrieval Examples. 5-006
5.2.2 VMS ACCOUNTING Image Termination File (DEAIMG) 5-006
5.2.2.1 DEAIMG File Organization 5-007
5.2.2.2 DEAIMG Data Elements List. 5-007
5.2.2.3 DEAIMG Usage Considerations. 5-008
5.2.2.4 DEAIMG Retrieval Examples. 5-009
5.2.3 VMS ACCOUNTING LOGIN Failed File (DEA_LF). . . 5-009
5.2.3.1 DEA_LF File Organization 5-010
5.2.3.2 DEA_LF Data Elements List. 5-010
5.2.3.3 DEA_LF Usage Considerations. 5-011
5.2.3.4 DEA_LF Retrieval Examples. 5-012
5.2.4 VMS ACCOUNTING Print Queued File (DEAPRQ). . . 5-012
5.2.4.1 DEAPRQ File Organization 5-013
5.2.4.2 DEAPRQ Data Elements List. 5-013
5.2.4.3 DEAPRQ Usage Considerations. 5-014
5.2.4.4 DEAPRQ Retrieval Examples. 5-015
5.2.5 VMS ACCOUNTING Initialization File (DEAINT). . 5-015
5.2.5.1 DEAINT File Organization 5-016
5.2.5.2 DEAINT Data Elements List. 5-016
5.2.5.3 DEAINT Usage Considerations. 5-017
5.2.5.4 DEAINT Retrieval Example 5-018

5.3 VAX/VMS Monitor Information Area Files 5-019
5.3.1 VMS MONITOR Process File (DEMPRO). 5-019
5.3.1.1 DEMPRO File Organization 5-020
5.3.1.2 DEMPRO Data Elements List. 5-020
5.3.1.3 DEMPRO Usage Considerations. 5-021
5.3.1.4 DEMPRO Retrieval Examples. 5-022
5.3.2 VMS MONITOR Modes File (DEMMOD). 5-022
5.3.2.1 DEMMOD File Organization 5-023
5.3.2.2 DEMMOD Data Elements List. 5-023
5.3.2.3 DEMMOD Usage Considerations. 5-024
5.3.2.4 DEMMOD Retrieval Examples. 5-025
5.3.3 VMS MONITOR Disk File (DEMDSK) 5-025
5.3.3.1 DEMDSK File Organization 5-026
5.3.3.2 DEMDSK Data Elements List. 5-026
5.3.3.3 DEMDSK Usage Considerations. 5-027
5.3.3.4 DEMDSK Retrieval Examples. 5-028
5.3.4 VMS MONITOR SCS File (DEMSCS). 5-028
5.3.4.1 DEMSCS File Organization 5-029
5.3.4.2 DEMSCS Data Elements List. 5-029
5.3.4.3 DEMSCS Usage Considerations. 5-030
5.3.4.4 DEMSCS Retrieval Examples. 5-031
5.3.5 VMS MONITOR RMS File (DEMRMS). 5-031
5.3.5.1 DEMRMS File Organization 5-032
5.3.5.2 DEMRMS Data Elements List. 5-032
5.3.5.3 DEMRMS Usage Considerations. 5-034
5.3.5.4 DEMRMS Retrieval Examples. 5-034
5.3.6 VMS System Profile File (DEMSPR) 5-035
5.3.6.1 DEMSPR File Organization 5-036
5.3.6.2 DEMSPR Data Elements List. 5-036
5.3.6.3 DEMSPR Usage Considerations. 5-040
5.3.6.4 DEMSPR Retrieval Examples. 5-041

5.4 VAX/VMS System Usage Information Area Files. . . 5-045
5.4.1 VMS Disk Device File (DESDKD). 5-045
5.4.1.1 DESDKD File Organization 5-046
5.4.1.2 DESDKD Data Elements List. 5-046
5.4.1.3 DESDKD Usage Considerations. 5-047
5.4.1.4 DESDKD Retrieval Examples. 5-048
5.4.2 VMS Disk Quota File (DESDKQ) 5-048
5.4.2.1 DESDKQ File Organization 5-049

5.4.2.2 DESDKQ Data Elements List. 5-049
5.4.2.3 DESDKQ Usage Considerations. 5-050
5.4.2.4 DESDKQ Retrieval Examples. 5-051
5.4.3 VMS Disk Usage File (DESDKU) 5-051
5.4.3.1 DESDKU File Organization 5-052
5.4.3.2 DESDKU Data Elements List. 5-052
5.4.3.3 DESDKU Usage Considerations. 5-053
5.4.3.4 DESDKU Retrieval Examples. 5-054
5.4.4 VMS Process Activity File (DESPRX) 5-054
5.4.4.1 DESPRX File Organization 5-055
5.4.4.2 DESPRX Data Elements List. 5-055
5.4.4.3 DESPRX Usage Considerations. 5-056
5.4.4.4 DESPRX Retrieval Examples. 5-057
5.4.5 VMS System Status File (DESSYU). 5-057
5.4.5.1 DESSYU File Organization 5-058
5.4.5.2 DESSYU Data Elements List. 5-058
5.4.5.3 DESSYU Usage Considerations. 5-059
5.4.5.4 DESSYU Retrieval Examples. 5-060

Chapter 6. DATA SOURCES. 6-001

6.1 Data Collection and Transport to CA MICS 6-001
6.1.1 Data Collector Operations. 6-002
6.1.1.1 VMS ACCOUNTING Data Collection 6-003
6.1.1.2 VMS MONITOR Data Collection. 6-004
6.1.1.3 CA MICS VMS System Usage Data Collection (DEXSUS) 6-005
6.1.2 Data Volume Considerations 6-007
6.1.3 Transporting Data from VMS to MVS. 6-008
6.1.4 Errors Identified in Measurement Data. 6-008

6.2 VMS ACCOUNTING Record Descriptions 6-009
6.2.1 PROCESS TERMINATION Record 6-009
6.2.2 IMAGE TERMINATION Record 6-010
6.2.3 SYSTEM INITIALIZATION Record 6-010
6.2.4 LOGIN FAILED Record. 6-011
6.2.5 PRINT QUEUED Record. 6-011

6.3 VMS MONITOR Record Descriptions. 6-012
6.3.1 FILE HEADER Record 6-012
6.3.2 SYSTEM INFORMATION Record. 6-013
6.3.3 CLUSTER Record 6-013
6.3.4 DECNET Record. 6-014
6.3.5 DISK Record. 6-014
6.3.6 DLOCK Record 6-015
6.3.7 FCP Record 6-015
6.3.8 FILE SYSTEM CACHE Record 6-016
6.3.9 IO Record. 6-016
6.3.10 LOCK Record 6-017
6.3.11 MODES Record. 6-017
6.3.12 MSCP SERVER Record. 6-018
6.3.13 PAGE Record 6-018
6.3.14 POOL Record 6-019
6.3.15 PROCESSES Record. 6-019
6.3.16 RMS Record. 6-020
6.3.17 SCS Record. 6-020
6.3.18 STATES Record 6-021
6.3.19 SYSTEM Record 6-021
6.3.20 TRANSACTION Record. 6-022
6.3.21 VECTOR Record 6-022

6.4 CA MICS VMS System Usage (DEXSUS) Record Descriptions 6-023
6.4.1 DEXSUS FILE HEADER Record. 6-023
6.4.2 DEXSUS SYSTEM INFORMATION Records. 6-024
6.4.3 DEXSUS SYI1 Record 6-024
6.4.4 DEXSUS SYI2 Record 6-025
6.4.5 DEXSUS SYI3 Record 6-025
6.4.6 DEXSUS SYI4 Record 6-026
6.4.7 DEXSUS SYSTEM MEMORY 1 Records 6-026
6.4.8 DEXSUS SYSTEM MEMORY 2 Records 6-027
6.4.9 DEXSUS DISK DEVICE Records 6-027
6.4.10 DEXSUS DISK QUOTA Records 6-028
6.4.11 DEXSUS DISK USAGE Records 6-028
6.4.12 DEXSUS FILE TRAILER Record. 6-029

Chapter 7. DEFINING PARAMETERS 7-001

7.1 Environmental Considerations 7-001

7.2 Complex Level Parameters 7-002
7.2.1 Analyzer Definition Statements (DEXGENIN). . . 7-002
7.2.2 VAX Account Code Structure (DEXACCT) 7-003
7.2.3 Code the VAX/VMS Account Code Exit (DEXACRT) . 7-006
7.2.4 Code the VAX/VMS Disk Account Code Exit (DEXACRTD) 7-008

7.3 Unit Level Parameters. 7-011
7.3.1 VAX/VMS Processing Options (DEXOPS). 7-011
7.3.2 Input DD Statements for VAX/VMS (INPUTDEX) . . 7-035
7.3.3 Input Needed to Generate VAX System Code (DEXPGEN). 7-038
7.3.4 Database Space Modeling (DBMODEL). 7-038
7.3.4.1 Data Retention Specifications (FILE Statements). 7-039
7.3.4.2 DBMODEL Input Statements 7-041

7.4 Other Related Parameters 7-042

CHAPTER 8. INSTALLATION. 8-001

8.1 Installing the DEXSUS.COM Collection Routine . . 8-001

Chapter 9. PROCESSING. 9-001

9.1 Processing Considerations. 9-001

9.2 Daily Update Processing Flow 9-002
9.2.1 Input Raw Data 9-002
9.2.2 Information Area Processing. 9-003
9.2.3 Database Timespan Processing 9-004
9.2.4 File Aging 9-004

9.3 User Exits 9-005

9.4 Using COMMON Input DDnames and DEXINPUT exits. . 9-006

Chapter 10. MODIFICATION. 1-001

10.1 Parameter Modification. 1-001
10.1.1 Change the Number of Work Files for DAY074. . 1-002
10.1.2 Enable Internal Step Restart. 1-003
10.1.3 Implement Incremental Update. 1-004

APPENDIX A. MESSAGES A-001

APPENDIX B. DATA DICTIONARY. B-001

B.1 Common Data Elements B-001
B.1.1 CA MICS Common Elements. B-002
DAY Day Of The Month. B-002
DAYNAME Name of Day of Week B-002
DURATION Length of Recording Interval. B-002
ENDTS End Time Stamp. B-006
HOUR Hour Of The Day B-009
INTERVLS Number of Recording Intervals B-010
MICSVER CA MICS Version Number. B-010
MONTH Month Of The Year B-010
ORGSYSID Originating System Identification B-011
STARTTS Start Time Stamp. B-012
SYSID System Identifier B-015
WEEK Week Of The Year. B-016
YEAR YEAR Of The Century B-016
ZONE Time Zone B-017
B.1.2 DEX Common Elements. B-018
DEXACCNT Account Name. B-018
DEXBOOTS Monitor System Boot Time Stamp. B-018
DEXDEVIC Device Name B-018
DEXDUR Requested Recording Interval Time B-018
DEXGRP User Identification Code Group. B-019

D E T A I L T A B L E O F C O N T E N T S PAGE

DEXINTNO Reserved Field Incremented by Monitor . . B-019
DEXJOB Jobname B-019
DEXJOBID Job ID. B-020
DEXMBR User Identification Code Member B-020
DEXMONVR VMS Monitor Version B-020
DEXNADDR Node Address. B-021
DEXNODE Nodename. B-021
DEXOWNID Owner ID. B-021
DEXPID Process ID. B-022
DEXPRIV1 Privilege 1 Mask. B-022
DEXPRIV2 Privilege 2 Mask. B-022
DEXPRTY Priority. B-023
DEXQUEUE Queue Name. B-023
DEXRMTID Remote ID B-023
DEXSTATS Status. B-023
DEXSUSTB DEXSUS System Boot Time Stamp B-024
DEXSUSTS DEXSUS Collect Time Stamp B-024
DEXTERM Terminal Name B-024
DEXTS Monitor Time Stamp. B-025
DEXUSER Username. B-025
DEXVOLNM Volume Name B-025
DEXVOLSR Volume Serial Number. B-026

B.2 VAX/VMS Accounting Information (DEA) Area. . . . B-027
B.2.1 File Relationships and Common Data Elements. . B-027
B.2.2 The VMS Accounting Process Termination (DEAPRC) File. B-028
PRCBUFIO Buffered I/Os B-029
PRCCOST Processing Charges. B-029
PRCCPUNI Number of Instructions Executed B-030
PRCCPUTM Total Process CPU Time. B-030
PRCDIRIO Direct I/Os B-030
PRCFLTIO Page Fault I/Os B-031
PRCIMAGE Image Count B-031
PRCPAGFL Page File Peak Usage. B-031
PRCPAGFT Page Faults B-032
PRCPRCNT Process Count B-032
PRCTYPE Process Type. B-032
PRCVOLS Volumes Mounted B-032
PRCWKSPK Working Set Peak. B-033
B.2.3 The VMS Accounting Image Termination (DEAIMG) File B-034
IMGBUFIO Buffered I/Os B-035
IMGCOST Processing Charges. B-035
IMGCPUNI Number of Instructions Executed B-036
IMGCPUTM Total Image CPU Time. B-036
IMGDIRIO Direct I/Os B-036
IMGFLTIO Page Fault I/Os B-037
IMGIMAGE Image Sequence Number B-037
IMGIMCNT Image Count B-037

IMGNAME Image Name. B-037
IMGPAGFL Page File Peak Usage. B-038
IMGPAGFT Page Faults B-038
IMGTYPE Process Type. B-039
IMGVOLS Volumes Mounted B-039
IMGWKSPK Working Set Peak. B-039
B.2.4 The VMS Accounting LOGIN Failed (DEA_LF) File. B-040
B.2.5 The VMS Accounting Print Queued (DEAPRQ) File. B-042
PRQCOST Processing Charges. B-043
PRQGETS GET Count B-043
PRQPAGES Print Pages B-043
PRQQIOS QIO Count B-044
PRQQUETM Queue Time. B-044
PRQQUETS Queue Time Stamp. B-044
PRQSYMTM Symbiont Time B-045
B.2.6 The VMS Accounting Initialization (DEAINT) File B-046
INTBUFIO Buffered I/Os B-047
INTCPUNI Number of Instructions Executed B-047
INTCPUTM CPU Time. B-048
INTDIRIO Direct I/Os B-048
INTFLTIO Page Fault I/Os B-048
INTIMAGE Image Count B-048
INTPAGFL Page File Peak Usage. B-049
INTPAGFT Page Faults B-049
INTVOLS Volumes Mounted B-049
INTWKSPK Working Set Peak. B-049
B.3 VAX/VMS Monitor Information (DEM) Area B-050
B.3.1 File Relationships and Common Data Elements. . B-050
B.3.2 The VMS Monitor Process (DEMPRO) File. B-051
PROBUFIO Buffered I/Os B-052
PROCPUNI Number of Instructions Executed B-052
PROCPUTM CPU Time. B-053
PRODIRIO Direct I/Os B-053
PROEID Extended Process ID B-053
PROEVNTF Event Flag Wait Mask. B-054
PROGLBPG Global Page Count B-054
PROGRP User Identification Code Group. B-054
PROMBR User Identification Code Member B-054
PROMXGPG Maximum Global Page Count B-055
PROMXPPG Maximum Process Page Count. B-055
PROPGFLT Page Faults B-055
PROPID Process ID. B-056
PROPNAME Process Name. B-056
PROPRCPG Process Page Count. B-056
PROPRTY Priority. B-056
PROSTATE State B-057
PROSTATF Status Flags. B-057
B.3.3 The VMS Monitor Modes (DEMMOD) File. B-059

DETAIL TABLE OF CONTENTS PAGE

MODCOMTM Compatibility Mode Time B-060
MODCPU CPU Identifier. B-060
MODCPUNI Number of Instructions Executed B-061
MODCPUTM Total CPU Time. B-061
MODEXETM Executive Mode Time B-061
MODIDLTM Idle Time B-062
MODINTTM Interrupt Stack Time. B-062
MODKERTM Kernel Mode Time. B-062
MODMPSTM Multi-Processor Sync Time B-063
MODPCCOM Compatibility Mode Percent. B-063
MODPCCPU Total CPU Percent B-063
MODPCEXE Executive Mode Percent. B-064
MODPCIDL Idle Percent. B-064
MODPCINT Interrupt Stack Percent B-064
MODPCKER Kernel Mode Percent B-065
MODPCMPS Multi-Processor Sync Percent. B-065
MODPCSUP Supervisor Mode Percent B-066
MODPCUSR User Mode Percent B-066
MODSUPTM Supervisor Mode Time. B-066
MODUSRTM User Mode Time. B-067
MODVECTM CPU Time of Vector Consumers. B-067
B.3.4 The VMS Monitor Disk (DEMDSK) File B-068
DSKALLOC Allocation Class. B-069
DSKAVRES Disk Average Response Time. B-069
DSKCTRLR Disk Controller B-070
DSKFLAGS Flags B-070
DSKIOOPS Disk I/O Operations B-070
DSKIOQUE Disk I/O Queue Length Samples Sum B-070
DSKNODE Nodename. B-071
DSKPSIO Disk I/O Operation Rate B-071
DSKUNIT Disk Unit Number. B-071
DSKVOL Disk Volume Name. B-072
B.3.5 The VMS Monitor SCS (DEMSCS) File. B-073
SCSBTKBM Block Transfer Kilobytes Mapped B-074
SCSBTRD Block Transfer Request Datas. B-074
SCSBTSND Block Transfer Send Datas B-074
SCSCQBFR Connections Queued for Buffer Descript. . B-075
SCSCQSND Connections Queued for Send Credit. . . . B-075
SCSDGDIS Datagrams Discarded B-075
SCSDGRCV Datagrams Received. B-075
SCSDGSNT Datagrams Sent. B-076
SCSKBRCV Kilobytes Received by Request Datas . . . B-076
SCSKBSND Kilobytes Sent by Send Datas. B-076
SCSMSGRC Sequenced Messages Received B-076
SCSMSGSN Sequenced Messages Sent B-077
SCSNODE Node Identifier B-077
SCSPSKBT Block Transfer Kilobytes Mapped Rate. . . B-077
B.3.6 The VMS Monitor RMS (DEMRMS) File. B-079
RMSALAST File Append Lock Blocking ASTs. B-080
RMSALCNV File Append Lock Conversions. B-081

RMSALDEQ File Append Lock Dequeues B-081
RMSALENQ File Append Lock Enqueues B-081
RMSBUCKT File Bucket Splits. B-082
RMSCLOSE File Closes B-082
RMSCONN File Connects B-082
RMSDELET File Deletes. B-082
RMSDISCN File Disconnects. B-083
RMSEXTBL File Extend Blocks. B-083
RMSEXTEN File Extends. B-083
RMSFBAST File Lock Blocking ASTs B-083
RMSFILE File Number B-084
RMSFILNM File Name B-084
RMSFLUSH File Flushes. B-084
RMSGBAST File Global Buffer Blocking ASTs. B-085
RMSGBCNV File Global Buffer Conversions. B-085
RMSGBDEQ File Global Buffer Dequeues B-085
RMSGBENQ File Global Buffer Enqueues B-085
RMSGBRIO File Global Buffer Read I/Os. B-086
RMSGBWIO File Global Buffer Write I/Os B-086
RMSGCATT File Global Cache Attempts. B-086
RMSGCHIT File Global Cache Hits. B-087
RMSGETBT File GET Bytes. B-087
RMSGSCNV File Global Section Conversions B-087
RMSGSDEQ File Global Section Dequeues. B-088
RMSGSENQ File Global Section Enqueues. B-088
RMSKEYFD Keyed Finds B-088
RMSKEYGT Keyed GETs. B-088
RMSKEYPT Keyed PUTs. B-089
RMSLBAST File Local Buffer Blocking ASTs B-089
RMSLBCNV File Local Buffer Conversions B-089
RMSLBDEQ File Local Buffer Dequeues. B-090
RMSLBENQ File Local Buffer Enqueues. B-090
RMSLBRIO File Local Buffer Read I/Os B-090
RMSLBWIO File Local Buffer Write I/Os. B-090
RMSLCATT File Local Cache Attempts B-091
RMSLCHIT File Local Cache Hits B-091
RMSLKCNV File Lock Conversions B-091
RMSLKDEQ File Lock Dequeues. B-092
RMSLKENQ File Lock Enqueues. B-092
RMSMULTB File Multibucket Splits B-092
RMSOPEN File Opens. B-093
RMSORG Organization. B-093
RMSPUTBT File PUT Bytes. B-093
RMSRDBT File Read Bytes B-093
RMSREAD File Reads. B-094
RMSREWND File Rewinds. B-094
RMSRFAFD Record File Address Finds B-094
RMSRFAGT Record File Address GETs. B-094
RMSRLCNV Record Lock Conversions B-095
RMSRLDEQ Record Lock Dequeues. B-095
RMSRLENQ Record Lock Enqueues. B-095
RMSSEQFD Sequential Finds. B-096
RMSSEQGT Sequential GETs B-096
RMSSEQPT Sequential PUTs B-096
RMSTRUBL File Truncate Blocks. B-096
RMSTRUNC File Truncates. B-097
RMSUPDBT File Update Bytes B-097
RMSUPDTE File Updates. B-097
RMSWRBT File Write Bytes. B-097
RMSWRITE File Writes B-098
B.3.7 The VMS System Profile (DEMSPR) File B-099
SPRACCS File Access B-103
SPRALLOC Disk Allocations. B-103
SPRARLPK Arriving Local Packets. B-104
SPRARTPK Arriving Transit Packets. B-104
SPRBAI Blocking ASTs, Incoming B-104
SPRBAL Blocking ASTs, Local. B-104
SPRBAO Blocking ASTs, Outgoing B-105
SPRBIGHL Largest Block Unused. B-105
SPRBIOS Buffered I/Os B-105
SPRBITAT Storage Bitmap Cache Attempts B-106
SPRBITHT Storage Bitmap Cache Hits B-106
SPRBLAST Blocking ASTs B-106
SPRBUFFL Receiver Buffer Failures. B-106
SPRBUFIO Buffered I/Os B-107
SPRBWAIT MSCP Buffer Waits B-107
SPRCALLS FCP Calls B-107
SPRCEF Common Event Flag Wait. B-108
SPRCENQ Converted Enqueues. B-108
SPRCLSCT CLS CLUSTER Record Count (Type 19). . . . B-108
SPRCLSDR CLS CLUSTER Record Duration (Type 19) . . B-108
SPRCLSTM CLS CPU Busy Time B-109
SPRCNTAB Transactions Aborted Count. B-109
SPRCNTAD Add Remote Branch Ops Count B-109
SPRCNTBR Start Remote Branch Ops Count B-110
SPRCNTCO Transactions Committed Count. B-110
SPRCNTD1 Transactions < 1 Second Count B-110
SPRCNTD2 Transactions 1-2 Seconds Count. B-110
SPRCNTD3 Transactions 2-3 Seconds Count. B-111
SPRCNTD4 Transactions 3-4 Seconds Count. B-111
SPRCNTD5 Transactions 4-5 Seconds Count. B-111
SPRCNTD6 Transactions > 5 Seconds Count. B-111
SPRCNTEN Transactions Ended Count. B-112
SPRCNTOP One-Phase Commits Initiated B-112
SPRCNTPR Transactions Prepared Count B-112
SPRCNTST Transactions Started Count. B-112
SPRCOLPG Collided Page Wait. B-113
SPRCOM Compute State Wait, Inswapped B-113
SPRCOMO Compute State Wait, Outswapped. B-113
SPRCPUNI Number of Instructions Executed B-114

DETAIL TABLE OF CONTENTS		PAGE
SPRCPUTM	System CPU Busy Time.	B-114
SPRCUR	Current Processes	B-115
SPRCURLK	Current Locks	B-115
SPRCURRS	Current Resources	B-115
SPRDATAT	Directory Data Cache Attempts	B-115
SPRDATHT	Directory Data Cache Hits	B-116
SPRDEADF	Deadlocks Found	B-116
SPRDEADS	Deadlock Searches	B-116
SPRDEQ	Dequeues.	B-117
SPRDIOS	Direct I/Os	B-117
SPRDIRIO	Direct I/Os	B-117
SPRDLKCT	DLK DLOCK Record Count (Type 14).	B-118
SPRDLKDR	DLK DLOCK Record Duration (Type 14) . . .	B-118
SPRDLM	Deadlock Messages	B-118
SPRDMZFL	Demand Zero Page Faults	B-118
SPRDNTCT	DNT DECNET Record Count (Type 8).	B-119
SPRDNTDR	DNT DECNET Record Duration (Type 8) . . .	B-119
SPRDPLPK	Departing Local Packets	B-119
SPRDQI	Dequeue Incoming.	B-120
SPRDQL	Dequeue Local	B-120
SPRDQO	Dequeue Outgoing.	B-120
SPRECI	Enqueue Conversions Incoming.	B-120
SPRECL	Enqueue Conversions Local	B-121
SPRECO	Enqueue Conversions Outgoing.	B-121
SPRENQNQ	Enqueues Not Queued	B-121
SPRENQWT	Enqueue Waits	B-121
SPRERASE	Erase I/O Operations.	B-122
SPREXTAT	Extent Cache Attempts	B-122
SPREXTHT	Extent Cache Hits	B-122
SPRFCBAT	Directory FCB Cache Attempts.	B-123
SPRFCBHT	Directory FCB Cache Hits.	B-123
SPRFCPCT	FCP Record Count (Type 5)	B-123
SPRFCPDR	FCP Record Duration (Type 5).	B-124
SPRFCPFL	FCP Page Faults	B-124
SPRFCPRI	FCP Disk Read I/Os.	B-124
SPRFCPTM	FCP CPU Time.	B-125
SPRFCPWI	FCP Disk Write I/Os	B-125
SPRFHDAT	File Header Cache Attempts.	B-125
SPRFHDHT	File Header Cache Hits.	B-125
SPRFIDAT	File ID Cache Attempts.	B-126
SPRFIDHT	File ID Cache Hits.	B-126
SPRFILOP	Files Opened.	B-126
SPRFNI	Directory Functions Incoming.	B-127
SPRFNO	Directory Functions Outgoing.	B-127
SPRFPGFL	Free List Page Faults	B-127
SPRFPGWT	Free Page Wait.	B-128
SPRFRE	Free Page List Size	B-128
SPRFREEP	IO Free Page Count.	B-129
SPRFRGMT	MSCP Fragments.	B-129
SPRFRPG	Free Page Count	B-129

DETAIL TABLE OF CONTENTS		PAGE
SPRFRPGC	PGE Free Page Count	B-130
SPRFSCCT	FSC CACHE Record Count (Type 11).	B-130
SPRFSCDR	FSC CACHE Record Duration (Type 11) . . .	B-131
SPRGBLFL	Global Valid Page Faults.	B-131
SPRHIB	Hibernate Wait, Inswapped	B-131
SPRHIBO	Hibernate Wait, Outswapped.	B-132
SPRHOLES	Nonpaged Pool Unused KBytes	B-132
SPRINSWP	Inswaps	B-132
SPRIOVCT	IOV IO Record Count (Type 4).	B-133
SPRIOVDR	IOV IO Record Duration (Type 4)	B-133
SPRIOVFL	IO Page Faults.	B-133
SPRIPFRE	Intermediate Request Packets Free	B-134
SPRIPUSE	Intermediate Request Packets in Use . . .	B-134
SPRKBUSE	Nonpaged Pool KBytes in Use	B-134
SPRLCI	Lock Conversions, Incoming.	B-134
SPRLCL	Lock Conversions, Local	B-135
SPRLCO	Lock Conversions, Outgoing.	B-135
SPRLEF	Local Event Flag Wait, Inswapped.	B-135
SPRLEFO	Local Event Flag Wait, Outswapped	B-135
SPRLOGNM	Logical Name Translations	B-136
SPRLOK	Total All Locks	B-136
SPRLOKCT	LOK LOCK Record Count (Type 7).	B-137
SPRLOKDR	LOK LOCK Record Duration (Type 7)	B-137
SPRLPFRE	Large Request Packets Free.	B-137
SPRLPUSE	Large Request Packets in Use.	B-137
SPRLRGPK	Large Request Packets Left.	B-138
SPRMAILW	Mailbox Writes.	B-138
SPRMASK	Record Construction Audit Mask.	B-138
SPRMDPG	Modified Page Count	B-139
SPRMDPGC	PGE Modified Page Count	B-139
SPRMNFP	Minimum Free Page Count	B-140
SPRMNFRE	Minimum Free Page List Size	B-140
SPRMNHOL	Minimum Unused KBytes	B-140
SPRMNIFP	IO Minimum Free Page Count.	B-141
SPRMNIMP	IO Minimum Modified Page Count.	B-141
SPRMNIPF	Minimum Intermediate Req Packets Free . .	B-141
SPRMNLBK	Minimum Largest Block	B-141
SPRMNLPF	Minimum Large Request Packets Free. . . .	B-142
SPRMNLPK	Minimum Large Request Packets Left. . . .	B-142
SPRMNMP	Minimum Modified Page Count	B-142
SPRMNPFP	PGE Minimum Free Page Count	B-143
SPRMNPMP	PGE Minimum Modified Page Count	B-143
SPRMNSBK	Minimum Smallest Block.	B-143
SPRMNSPF	Minimum Small Request Packets Free. . . .	B-144
SPRMNUCN	Minimum Unused Contiguous Space	B-144
SPRMODP	IO Modified Page Count.	B-144
SPRMPGFL	Modified List Page Faults	B-144
SPRMSCCT	MSC MSCP Server Record Count (Type 21). .	B-145
SPRMSCDR	MSC MSCP Server Record Duration(Type 21).	B-145
SPRMSCRD	MSCP Reads.	B-145

SPRMSCSP MSCP Splits B-146
SPRMSCWR MSCP Writes B-146
SPRMSTAB Max Transaction Abort Rate. B-146
SPRMSTAD Max Transaction Remote Add Rate B-147
SPRMSTBR Max Transaction Remote Start Rate B-147
SPRMSTCO Max Transaction Total Commit Rate B-147
SPRMSTEN Max Transaction End Rate. B-148
SPRMSTOP Max Transaction One-Phase Init Rate . . . B-148
SPRMSTPR Max Transaction Prepare Rate. B-148
SPRMSTST Max Transaction Start Rate. B-149
SPRMWAIT Miscellaneous Resource Wait B-149
SPRMXCEF Maximum Common Event Flag Wait. B-149
SPRMXCLK Maximum Current Locks B-149
SPRMXCP Maximum Current Processes B-150
SPRMXCPW Maximum Collided Page Wait. B-150
SPRMXCRS Maximum Current Resources B-150
SPRMXCSI Maximum Compute State Wait, Inswapped . . B-151
SPRMXCSO Maximum Compute State Wait, Outswapped. . B-151
SPRMXFPW Maximum Free Page Wait. B-151
SPRMXHWI Maximum Hibernate Wait, Inswapped B-152
SPRMXHWO Maximum Hibernate Wait, Outswapped. . . . B-152
SPRMXIPU Maximum Intermediate Req Packet in Use. . B-152
SPRMXKBU Maximum Nonpaged Pool KBytes in Use . . . B-153
SPRMXLEI Maximum Local Event Flag Wait, Inswap . . B-153
SPRMXLEO Maximum Local Event Flag Wait, Outswap. . B-153
SPRMXLPU Maximum Large Request Packets in Use. . . B-154
SPRMXMRW Maximum Miscellaneous Resource Wait . . . B-154
SPRMXOST Maximum Other States. B-154
SPRMXPFW Maximum Page Fault Wait B-155
SPRMXPRC Maximum Process Count B-155
SPRMXSMB Maximum Block Less Than or Equal to 32. . B-155
SPRMXSPI Maximum Suspended Wait, Inswapped B-156
SPRMXSPO Maximum Suspended Wait, Outswapped. . . . B-156
SPRMXSPU Maximum Small Request Packets in Use. . . B-156
SPRNEI New Enqueue Incoming. B-157
SPRNEL New Enqueue Local B-157
SPRNENQ New Enqueues. B-157
SPRNEO New Enqueue Outgoing. B-157
SPRNEWFL New Files B-158
SPRNLI New Locks, Incoming B-158
SPRNLL New Locks, Local. B-158
SPRNLO New Locks, Outgoing B-158
SPROPEN Files Opened. B-159
SPROSTAT Other States. B-159
SPRPCBIT Storage Bitmap Cache Hit Percentage . . . B-159
SPRPCDDC Directory Data Cache Hit Percentage . . . B-160
SPRPCEXT Extent Cache Hit Percentage B-160
SPRPCFCB Directory FCB Cache Hit Percentage. . . . B-161
SPRPCFHC File Header Cache Hit Percentage. B-161
SPRPCFID File ID Cache Hit Percentage. B-161

DETAIL TABLE OF CONTENTS PAGE

SPRPCQOT Quota Cache Hit Percentage. B-162
SPRPFW Page Fault Wait B-162
SPRPGECT PGE PAGE Record Count (Type 3). B-162
SPRPGEDR PGE PAGE Record Duration (Type 3) B-163
SPRPGEFL PGE Page Faults B-163
SPRPGERD PGE Page Reads. B-164
SPRPGERI PGE Page Read I/Os. B-164
SPRPGEWI PGE Page Write I/Os B-164
SPRPGEWR PGE Page Writes B-164
SPRPGFLT Page Faults B-165
SPRPGRD Page Reads. B-165
SPRPGWT Page Writes B-166
SPRPKLOS Transit Packets Lost. B-166
SPRPOLCT POL POOL Record Count (Type 6). B-166
SPRPOLDR POL POOL Record Duration (Type 6) B-166
SPRPRCNT Process Count B-167
SPRPRDIO Page Read I/Os. B-167
SPRPSACC File Access Rate. B-167
SPRPSALL Disk Allocation Rate. B-168
SPRPSARL Arriving Local Packet Rate. B-168
SPRPSAST Blocking ASTs Rate. B-169
SPRPSATP Arriving Transit Packet Rate. B-169
SPRPSBAI Blocking ASTs Incoming Rate B-169
SPRPSBAL Blocking ASTs Local Rate. B-170
SPRPSBAO Blocking ASTs Outgoing Rate B-170
SPRPSBIO Buffered I/O Rate B-170
SPRPSBIT Storage Bitmap Cache Attempt Rate B-171
SPRPSBUF Buffered I/O Rate B-171
SPRPSCAL FCP Call Rate B-171
SPRPSCEQ Converted Enqueue Rate. B-172
SPRPSDDC Directory Data Cache Attempt Rate B-172
SPRPSDEQ Dequeue Rate. B-172
SPRPSDFD Deadlocks Found Rate. B-173
SPRPSDIO Direct I/O Rate B-173
SPRPSDIR Direct I/O Rate B-173
SPRPSDLM Deadlock Message Rate B-174
SPRPSDMZ Demand Zero Page Fault Rate B-174
SPRPSDPL Departing Local Packet Rate B-174
SPRPSDQI Dequeue Incoming Rate B-175
SPRPSDQL Dequeue Local Rate. B-175
SPRPSDQO Dequeue Outgoing Rate B-175
SPRPSDSR Deadlock Search Rate. B-176
SPRPSECI Enqueue Conversion Incoming Rate. B-176
SPRPSECL Enqueue Conversion Local Rate B-176
SPRPSECO Enqueue Conversion Outgoing Rate. B-177
SPRPSEIO Erase I/O Rate. B-177
SPRPSENQ Enqueues Not Queued Rate. B-177
SPRPSEWT Enqueue Waits Rate. B-178
SPRPSEXT Extent Cache Attempt Rate B-178
SPRPSFCB Directory FCB Cache Attempt Rate. B-178

SPRPSFFL FCP Page Fault Rate B-179
SPRPSFHC File Header Cache Attempt Rate. B-179
SPRPSFID File ID Cache Attempt Rate. B-179
SPRPSFLT Page Fault Rate B-180
SPRPSFNI Directory Functions Incoming Rate B-180
SPRPSFNO Directory Functions Outgoing Rate B-181
SPRPSFOP Files Opened Rate B-181
SPRPSFPL Free List Page Fault Rate B-181
SPRPSFRI FCP Disk Read I/O Rate. B-182
SPRPSFWI FCP Disk Write I/O Rate B-182
SPRPSGBL Global Valid Page Fault Rate. B-183
SPRPSIFL IO Page Fault Rate. B-183
SPRPSINS Inswap Rate B-184
SPRPSLCI Lock Conversion Incoming Rate B-184
SPRPSLCL Lock Conversion Local Rate. B-184
SPRPSLCO Lock Conversion Outgoing Rate B-185
SPRPSLNT Logical Name Translation Rate B-185
SPRPSLOS Transit Packet Loss Rate. B-185
SPRPSMBX Mailbox Write Rate. B-186
SPRPSMPL Modified List Page Fault Rate B-186
SPRPSNEI New Enqueue Incoming Rate B-187
SPRPSNEL New Enqueue Local Rate. B-187
SPRPSNEO New Enqueue Outgoing Rate B-187
SPRPSNEQ New Enqueue Rate. B-187
SPRPSNFL New Files Rate. B-188
SPRPSNLI New Locks Incoming Rate B-188
SPRPSNLL New Locks Local Rate. B-189
SPRPSNLO New Locks Outgoing Rate B-189
SPRPSOPN File Open Rate. B-189
SPRPSPFL PGE Page Fault Rate B-190
SPRPSPGR Page Read Rate. B-190
SPRPSPGW Page Write Rate B-190
SPRPSPR Page Read I/O Rate. B-191
SPRPSPRI PGE Page Read I/O Rate. B-191
SPRPSPW Page Write I/O Rate B-192
SPRPSPWI PGE Page Write I/O Rate B-192
SPRPSQOT Quota Cache Attempt Rate. B-192
SPRPSRBF Receiver Buffer Failure Rate. B-193
SPRPSRED PGE Page Read Rate. B-193
SPRPSRIO Read I/O Rate B-193
SPRPSSPL Split Transfer Rate B-194
SPRPSSYS System Page Fault Rate. B-194
SPRPSTAB Transaction Abort Rate. B-195
SPRPSTAD Transaction Remote Add Rate B-195
SPRPSTBR Transaction Remote Start Rate B-195
SPRPSTCO Transaction Total Commit Rate B-195
SPRPSTEN Transaction End Rate. B-196
SPRPSTOP Transaction One-Phase Init Rate B-196
SPRPSTPR Transaction Prepare Rate. B-196
SPRPSTRN Window Turn Rate. B-197
SPRPSTST Transaction Start Rate. B-197
SPRPSUNI Unlock Incoming Rate. B-197
SPRPSUNL Unlock Local Rate B-198
SPRPSUNO Unlock Outgoing Rate. B-198
SPRPSVLK Volume Lock Wait Rate B-198
SPRPSWIP Write In Progress Page Fault Rate B-199
SPRPSWRT PGE Page Write Rate B-199
SPRPWIO Page Write I/Os B-199
SPRQUOAT Quota Cache Attempts. B-200
SPRQUOHT Quota Cache Hits. B-200
SPRRDIOS Page Read I/Os. B-200
SPRREQST MSCP Requests B-200
SPRSIZE1 1 Block I/Os. B-201
SPRSIZE2 2-3 Block I/Os. B-201
SPRSIZE3 4-7 Block I/Os. B-201
SPRSIZE4 8-15 Block I/Os B-202
SPRSIZE5 16-31 Block I/Os. B-202
SPRSIZE6 32-63 Block I/Os. B-202
SPRSIZE7 64 And Larger Block I/Os. B-202
SPRSMBLK Block Less Than or Equal to 32. . . . . B-203
SPRSMHL Smallest Block Unused B-203
SPRSPFRE Small Request Packets Free. B-203
SPRSPLIT Split Transfers B-204
SPRSPUSE Small Request Packets in Use. B-204
SPRSTACT STA STATES Record Count (Type 1). B-204
SPRSTADR STA STATES Record Duration (Type 1) . . . B-204
SPRSUSP Suspended Wait, Inswapped B-205
SPRSUSPO Suspended Wait, Outswapped. B-205
SPRSYSFL System Page Faults. B-205
SPRSYTCT SYT SYSTEM Record Count (Type 17) B-206
SPRSYTDR SYT SYSTEM Record Duration (Type 17). . . B-206
SPRTRNCT TRN TRANS Record Count (Type 22). B-206
SPRTRNDR TRN TRANS Record Duration (Type 22) . . . B-207
SPRTRNGE DEXTS Range From Input Records. B-207
SPRTURN Window Turns. B-207
SPRUNCON Nonpaged Pool Unused Contiguous Space . . B-208
SPRUNI Unlocks, Incoming B-208
SPRUNL Unlocks, Local. B-208
SPRUNO Unlocks, Outgoing B-209
SPRVLOK Volume Lock Waits B-209
SPRWIPFL Write In Progress Page Faults B-209
SPRWTURN Window Turns. B-209

B.4 VAX/VMS System Usage Information (DES) Area. . . B-210
B.4.1 File Relationships and Common Data Elements. . B-211
B.4.2 The VMS Disk Device (DESDKD) File. B-212
DKDBLKSF Device Free Blocks. B-213
DKDBLKST Device Total Blocks B-213
DKDDEVTY Device Type B-213
DKDERRCT Device Error Count. B-213

DETAIL TABLE OF CONTENTS	PAGE
DKDVOFMT Volume Format Type.	B-214
DKDVOWNM Volume Owner Name	B-214
DKDVSNAM Volume Set Name	B-214
B.4.3 The VMS Disk Quota (DESDKQ) File	B-215
DKQMBHUS Disk Quota Usage Megabyte (2**20) Hrs . .	B-216
DKQOVLIM Disk Quota Overdraft Limit (Blocks) . . .	B-216
DKQPQUOT Disk Quota Permanent Quota (Blocks) . . .	B-216
DKQUIC Disk Quota UIC.	B-217
DKQUSAGE Disk Quota Usage (Blocks)	B-217
B.4.4 The VMS Disk Usage (DESDKU) File	B-218
DKUBLKSA Number of Blocks (512) Allocated.	B-219
DKUBLKST Number of Blocks (512) Total.	B-219
DKUBLKSU Number of Blocks (512) Used	B-219
DKUCNT File Count.	B-220
DKUCOST Disk Usage Charges.	B-220
DKUDIR Directory Name.	B-220
DKUDIRTL Directory Top Level	B-220
DKUFILE File Specification.	B-221
DKUMBHA Megabyte (2**20) Hours Allocated.	B-221
DKUMBHU Megabyte (2**20) Hours Used	B-222
B.4.5 The VMS Process Activity (DESPRX) File	B-223
PRXBUFIO Buffered I/Os	B-224
PRXCPUNI Number of Instructions Executed	B-224
PRXCPUTM CPU Time.	B-225
PRXDIRIO Direct I/Os	B-225
PRXEVNTF Event Flag Wait Mask.	B-225
PRXGLBPG Global Page Count	B-226
PRXGRP User Identification Code Group.	B-226
PRXIPID Internal Process ID	B-226
PRXMBR User Identification Code Member	B-226
PRXMXGPG Maximum Global Page Count	B-227
PRXMXPPG Maximum Process Page Count.	B-227
PRXPGFLT Page Faults	B-227
PRXPNAME Process Name.	B-228
PRXPRCPG Process Page Count.	B-228
PRXPRTY Priority.	B-228
PRXSTATE State	B-229
PRXSTATF Status Flags.	B-229
PRXTYPE Process Type.	B-230
PRXWKSET Process Working Set	B-230
B.4.6 The VMS System Status (DESSYU) File.	B-231
SYUBALSM Process Pages for Memory Allocation . . .	B-232
SYUCPUS Number of CPUs.	B-232
SYUHWNAM Hardware Name	B-232
SYUIRPC IRP Lookaside List, Current Packets . . .	B-233
SYUIRPI IRP Lookaside List, Initial Packets . . .	B-233
SYUIRPM IRP Lookaside List, Maximum Packets . . .	B-233
SYULRPC LRP Lookaside List, Current Packets . . .	B-233
SYULRPI LRP Lookaside List, Initial Packets . . .	B-234
SYULRPM LRP Lookaside List, Maximum Packets . . .	B-234

DETAIL TABLE OF CONTENTS	PAGE
SYUMEMAN Main Memory Mb (physical)	B-234
SYUMENDC Nonpaged Dynamic Memory, Current Size . .	B-234
SYUMENDI Nonpaged Dynamic Memory, Initial Size . .	B-235
SYUMEPDC Paged Dynamic Memory, Current Size. . . .	B-235
SYUMPWHI MPW_HILIMIT System Parameter.	B-235
SYUMXMEN Nonpaged Dynamic Memory, Maximum Size . .	B-235
SYUMXPRC MAXPROCESSCNT System Parameter.	B-236
SYUSRPC SRP Lookaside List, Current Packets . . .	B-236
SYUSRPI SRP Lookaside List, Initial Packets . . .	B-236
SYUSRPM SRP Lookaside List, Maximum Packets . . .	B-237
SYUUPTIM System Uptime	B-237
SYUVCPUS Number of Vector-Present Processors . . .	B-237
SYUVMSVR VMS Version	B-237

FIGURE TABLE OF CONTENTS

NUMBER	TITLE	PAGE
1-1	CA MICS VAX/VMS Analyzer Data Flow	1-001
3-101	Resource Intensive Processes - CPU & I/O .	3-005
3-102	Resource Intensive Processes - Images & Paging.	3-006
3-103	Process Terminations Status.	3-007
3-104	Weekly Process Name Summary.	3-008
3-105	Weekly Process Activity.	3-009
3-106	Monthly Process Activity	3-010
3-201	Disk Response Time	3-014
3-202	I/O Queue Length Average	3-015
3-203	Service Performed by the System.	3-016
3-204	MP Synchronization Time.	3-017
3-205	Executive Mode Time.	3-018
3-207	Hard Faults to Total	3-019
3-208	Available CPU Cycles	3-020
3-209	Overall Page Fault Rate.	3-021
3-210	Daily System Status - Uptime & Memory Use.	3-022
3-211	Daily System Summary - CPU Hourly Averages	3-023
3-212	Daily System Summary - Average Disk Device Statistics	3-024
3-213	Weekly Average Disk Response Time.	3-025
3-214	Service Performed by the System - CPU Avg.	3-026
3-301	Detail Directory Top Level Space	3-029
3-302	Detail Directory Top Level Ranking - Allocated	3-030
3-303	Detail Directory Top Level Ranking - Unused.	3-031
3-304	Detail Directory Structure and Space . . .	3-032
3-305	Detail File Space Ranking - Allocated. . .	3-033
3-306	Detail Disk Quota Statistics	3-034
3-307	Daily Directory Top Level Usage.	3-035
3-308	Daily Directory Top Level Ranking - Allocated	3-036
3-309	Daily Directory Top Level Ranking - Unused	3-037
3-310	Weekly Disk Device Free Space.	3-038
3-311	Weekly Disk Device Space Utilization . . .	3-039
3-312	Monthly Disk Usage Summary	3-040
3-313	Monthly Disk Usage Ranking - Allocated . .	3-041
3-314	Monthly Disk Usage Ranking - Unused. . . .	3-042
5-1	VAX/VMS Information Area Files	5-001
5-2	DEAPRC Time-Span Granularity Chart	5-004
5-3	DEAIMG Time-Span Granularity Chart	5-007
5-4	DEA_LF Time-Span Granularity Chart	5-010
5-5	DEAPRQ Time-Span Granularity Chart	5-013
5-6	DEAINT Time-Span Granularity Chart	5-016
5-7	DEMPRO Time-Span Granularity Chart	5-020
5-8	DEMMOD Time-Span Granularity Chart	5-023
5-9	DEMDSK Time-Span Granularity Chart	5-026

FIGURE TABLE OF CONTENTS

NUMBER	TITLE	PAGE
5-10	DEMSCS Time-Span Granularity Chart	5-029
5-11	DEMRMS Time-Span Granularity Chart	5-032
5-12	DEMSPR Time-Span Granularity Chart	5-036
5-13	DESDKD Time-Span Granularity Chart	5-046
5-14	DESDKQ Time-Span Granularity Chart	5-049
5-15	DESDKU Time-Span Granularity Chart	5-052
5-16	DESPRX Time-Span Granularity Chart	5-055
5-17	DESSYU Time-Span Granularity Chart	5-058
7-1	VAX/VMS Account Code Level Definition Worksheet	7-005
7-2	VAX/VMS Account Code Routine Definition Worksheet	7-007
7-2a	VAX/VMS Disk Account Code Routine Definition Worksheet.	7-010
7-3	VAX/VMS Analyzer Options Definition Worksheet	7-034
7-4	INPUTDEX JCL Definitions Worksheet	7-037
7-5	Members/Macros Needed/Updated by DEXPGEN Code.	7-038
7-6	Database Data Retention Definitions Worksheet	7-040
9-1a	Record/File Correspondence	9-002
9-1b	Record/File Correspondence	9-003
9-2	File/User Exit Correspondence.	9-005
9-3	File/Accounting Exit Correspondence. . . .	9-006

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 1

OVERVIEW

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 1

OVERVIEW

CHAPTER 1. OVERVIEW

As the number of management information systems installations using Hewlett Packard's (DEC) VAX/VMS systems continues to grow, the need to systematically plan, control, and evaluate the VMS environment is becoming increasingly important.

CA has responded to this need by developing the CA MICS VAX/VMS Analyzer, a data integration application that processes raw data for analyzing and managing VMS operating system information.

```
                                CA MICS       CA
  Data      CA MICS Data      Database   Management
 Source     Integration     Information    Support
              Product         Areas    Applications   Output
 _________   ________       _________
|   VMS   |  |        |    | _______ |   _______
|Utilities|  |        |    ||VAX/VMS||   |CA MICS |
|---------|  |CA MICS |    ||Acctg. ||   |Acctg. &|   _________
|Acctg.   |->|VAX/VMS |->||_______||->|Charge- |  |         |
|Monitor  |  |Analyzer|    | _______ |   |  back  |->| Reports |
|Anly/Disk|  |        |    ||VAX/VMS||   |_______|  |         |
|Diskquota|  |        |    ||Monitor||-------------->|         |
|Show     |  |        |    ||_______||   _______   |Inquiries|
|_________|  |________|    | _______ |   |CA MICS |  |         |
                           ||VAX/VMS||   |Capacity|->|_________|
                           ||Sys Usg||->|Planner |  |_________|
                           ||_______||   |_______|
                           |_________|
```

Figure 1-1. CA MICS VAX/VMS Analyzer Data Flow

The CA MICS VAX/VMS Analyzer provides the capability to display information on selected users, groups of users, hardware subsystems, and other categories. It also can be used to track system problems and assist in performing an in-depth audit of users of the various VMS facilities. An account code structure further enables VMS activity to be reported by any other relevant category.

The CA MICS VAX/VMS Analyzer processes data from VMS Accounting (VMS ACCOUNTING), VMS Monitor Utility (VMS MONITOR), and the CA MICS VMS System Usage collection routine (CA MICS DEXSUS). System Usage data collection includes the VMS Analyze/Disk_Structure Utility, the VMS SYSMAN Utility for DISKQUOTA SHOW information, and other information from VMS SHOW and Lexical functions. This combination of data sources contains system, file, and user activity information, making it an ideal data source in CA MICS for managing the VMS environment.

VMS ACCOUNTING provides utilization information on process and image resource usage, system initialization, system access, and print statistics. VMS MONITOR is a systems management tool that collects I/O statistics, page management statistics, CPU states and use, DECnet traffic, and other vital information in collection intervals. CA MICS DEXSUS is a collection routine that combines data from several VMS utilities and functions that measure disk space usage by volume, directory, and individual VMS files, disk quota information, plus a number of useful system parameter settings.

The data from VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS, when integrated into the CA MICS database, describes system performance data for managing system resources and utilization data for studying user activity and billing users for computer resource usage.

Because this data is stored in the CA MICS database in a format common to all CA MICS products, you can use standard CA MICS facilities to integrate CA MICS VAX/VMS Analyzer data with any other CA MICS management support applications such as financial management and capacity management. CA MICS offers online data access through the CA MICS Information Center Facility (MICF), which is a full-screen, menu-based processor that allows you to perform data analysis, inquiry, retrieval, and report functions.

The CA MICS VAX/VMS Analyzer integrates VMS performance and utilization data into the CA MICS database, thereby creating the CA MICS VAX/VMS Monitor, Accounting, and System Usage information areas that interface with:

- o CA MICS Accounting and Chargeback - The data provided to CA MICS Accounting and Chargeback supports accounting for consumed resources (e.g., CPU time), physical I/Os (e.g., the number of direct I/Os), disk space occupancy (e.g., megabyte hours of storage) and many other measures of resource utilization.

- o CA MICS Capacity Planner - The data provided to CA MICS Capacity Planner allows forecasting of VMS workloads and system utilizations. The Capacity Planner includes a standard applications for VAX/VMS CPU Planning.

This chapter provides you with a general discussion of the CA MICS VAX/VMS Analyzer in the following subsections:

1 - Primary Areas of Application
2 - Reporting and Inquiry Facilities
3 - Files Overview
4 - Product Prerequisites

1.1 - Primary Areas of Application

The CA MICS VAX/VMS Analyzer provides data and reporting capabilities that aid in managing the VMS environment in the following areas:

- Capacity Planning - The CA MICS VAX/VMS Analyzer maintains a database of historical VMS usage and performance data. This information can be used for growth/trend projections at both the system and organizational unit (e.g., division or department) levels.

- Performance Management - The CA MICS VAX/VMS Analyzer allows you to examine and analyze specific VMS workloads so you can optimize your hardware and software resources. This information can be used for monitoring system behavior on a routine basis, investigating user reports of degraded performance, performing system tuning, and assessing the affect of changes in system software and hardware configurations.

- Service Level Management - The CA MICS VAX/VMS Analyzer allows the user to use the data from VMS MONITOR and VMS ACCOUNTING to create and maintain Service Level Agreements (SLAs). Creating SLAs and reporting on the success of IS in meeting them requires that application data be in an easily accessible format. The CA MICS VAX/VMS Analyzer accomplishes this by combining data into user-specified groups (e.g., accounts payable) to provide consolidated reporting on SLA achievement. The CA MICS VAX/VMS Analyzer also allows you to monitor system downtime for reliability and for reporting availability.

- Management and Analytical Reporting - The CA MICS VAX/VMS Analyzer allows users to group and summarize information directly from the CA MICS database to create useful ad hoc management and analytical reports. CA MICS facilities allow resummarization when the data in the CA MICS database does not address specific reporting needs.

- Accounting and Chargeback - The CA MICS VAX/VMS Analyzer resource usage files contain data in a form that can be used for billing VMS system use. This data includes CPU time, I/O counts, disk space usage and other resource usage information. A standard interface to CA MICS Accounting and Chargeback permits accounting for VMS usage.

1.2 - Reporting and Inquiry Facilities

The CA MICS VAX/VMS Analyzer provides the capability to use the MICF and standard SAS language interface to create ad hoc reports. CA MICS provides these interactive capabilities that allow fast response to complex requests so that users have the requisite information to manage an IS organization on a daily basis. These capabilities are provided by:

- o MICF - MICF is a panel-oriented productivity tool that allows you to access information in the CA MICS database. When you define input, selection criteria, and report options, MICF dynamically builds an inquiry program that executes in either interactive or batch mode. Then, depending on your specific requirements, MICF either prints the inquiry report, displays the results at your terminal, or catalogs the results for later viewing.

- o The Standard SAS Language Interface - The CA MICS Workstation Facility (MWF) provides an online environment that supports the standard SAS language interface in both interactive and batch modes. SAS allows easy data manipulation and statistical analysis and provides coding facilities for programmers who are conducting extensive analysis efforts or designing new reports for IS business applications.

1.3 - Files Overview

The CA MICS VAX/VMS Analyzer builds the VAX/VMS Monitor Information Area, the VAX/VMS Accounting Information Area, and the VAX/VMS System Usage Information Area. Once created, these Information Areas maintain system and user workload and utilization data. This section defines the use and content of each file from the CA MICS VAX/VMS Analyzer information areas.

VAX/VMS ACCOUNTING FILES

The VAX/VMS Accounting Information Area maintains system and user utilization data for accounting and chargeback.

o Image Termination (DEAIMG) file

This file maintains information on the images terminated, including starting time, CPU time, page faults, fault I/O, direct and buffered I/O, maximum working set size, and volumes mounted. It is derived from the VMS ACCOUNTING type 3 record.

o Initialization (DEAINT) file

This file maintains information on system initializations. It is derived from the VMS ACCOUNTING type 5 record.

o LOGIN Failed (DEA_LF) file

This file maintains information on unsuccessful attempts to gain access to the system. It is derived from the VMS ACCOUNTING type 7 record.

o Print Queued (DEAPRQ) file

This file maintains information on printing, including start time, queue time, pages printed, and I/O count. It is derived from the VMS ACCOUNTING type 8 record.

o Process Termination (DEAPRC) file

This file maintains information on processes terminated, including starting time, CPU time, page faults, fault I/O, direct and buffered I/O, maximum working set size, and volumes mounted. It is derived from the VMS ACCOUNTING type 1 record.

VAX/VMS MONITOR FILES

The VAX/VMS Monitor Information Area maintains system workload and response data for monitoring system performance.

o Disk (DEMDSK) file

This file maintains information on VMS disk activity by volume, including number of I/O operations and queue lengths. It is derived from the VMS MONITOR type 12 record.

o Modes (DEMMOD) file

This file maintains information describing the time spent in each of the processor modes: interrupt, synchronization, kernel, executive, supervisor, user, compatibility, and idle. It is derived from the VMS MONITOR type 2 record.

Also, CPU time of Vector Consumers is provided from the VMS MONITOR type 23 record.

o Process (DEMPRO) file

This file maintains information on the processes executing in the system: process name, page count, I/Os, and CPU time. It is derived from the VMS MONITOR type 0 record.

o RMS (DEMRMS) file

This file maintains information on the VMS Record Management Services (RMS), including, among others, GET bytes, PUT bytes, update bytes, deletes, finds, connects, and disconnects. It is derived from the VMS MONITOR type 20 record.

o SCS (DEMSCS) file

This file maintains information on the VMS System Communication Services (SCS) activity, including datagrams sent, received, and discarded; send-datas; messages sent and received; and kilobytes transferred. It is derived from the VMS MONITOR type 15 record.

o System Profile (DEMSPR) file

This file maintains consolidated information on many aspects of system operation. It is derived from the VMS MONITOR class records for system class, including type 1, 3, 4, 5, 6, 7, 8, 11, 14, 17, 19, 21, and 22 records.

Specific types of information in this file:

- Overall system operation, including CPU busy, read I/Os, direct I/Os, buffered I/Os, free page count, modified page count, and number of active processes.
- Cluster-wide VMS lock activity, including enqueues, dequeues, and conversions.
- DECnet activity, including arriving and departing packets, lost packets, and buffer failures.
- Distributed Lock Management Facility, including enqueues, dequeues, and conversions.
- VMS file system ancillary control processes (ACPs), including new files, page faults, and read and write I/Os.
- VMS File System Cache, including cache attempts and hits for directories, file headers, file IDs, and extents.
- VMS I/O subsystem, including direct I/Os, buffered I/Os, page reads, page writes, and page faults.
- Lock management in VMS, including enqueues, dequeues, and deadlocks.
- MSCP server activities, including reads, writes, requests, and I/Os of various sizes.
- VMS Page Management System, including page faults, reads, read I/Os, writes, write I/Os, faults, and free pages.
- VMS space allocation in the nonpaged dynamic pool, including packets free and in use (small, intermediate, and large), kilobytes free and in use, and largest and smallest blocks available.
- Number of processes in each scheduler state: waiting, hibernating, suspended, inswapped, and outswapped.
- Transaction processing statistics for DECdtm services, including transaction starts, commits, aborts and ends.

VAX/VMS SYSTEM USAGE FILES

The VAX/VMS System Usage Information Area maintains disk usage data by user for accounting and chargeback, disk quota and disk device information for managing overall storage use,

process activity information based on combined Monitor and Accounting data for an enhanced view of Monitor data on performance by workload, and other useful information on system status.

o Disk Device (DESDKD) file

This file identifies device type, volume name, and volume set membership for each active disk device on each VMS system, and includes readings of device free blocks, device total blocks, and most recent error count. The source of this file is the VMS ANALYZE/DISK_STRUCTURE Utility type 01 record and the VMS SHOW DEVICE output.

o Disk Quota (DESDKQ) file

This file quantifies disk usage by disk device and Diskquota-displayed User ID Codes (UIC's). Measures include quota usage, permanent quota, overdraft limit, and megabyte-hours of quota usage storage. The source of this file is the VMS SYSMAN Utility, DISKQUOTA SHOW output.

o Disk Usage (DESDKU) file

This file quantifies disk usage by account code levels, plus directory top level at the DAYS time-span, and individual VMS directories and files at the DETAIL time-span. Measures include blocks allocated, blocks used, and megabyte-hours of storage. The source of this file is the VMS ANALYZE/DISK_STRUCTURE Utility type 02 record.

o Process Activity (DESPRX) file

This file quantifies the resource usage of each process on the VMS system during the monitoring interval, by user ID code (UIC), username, VMS account name, terminal name, and process type and name. The source of this file is the VMS ACCOUNTING Process Termination record, type 01, and the VMS MONITOR Processes record, type 00.

o System Status (DESSYU) file

This file provides a variety of system readings for each execution of the CA MICS DEXSUS collection routine, including system hardware name, system uptime, VMS version, number of CPU's, main memory, and many other system settings. The source of this file is the VMS MONITOR System Information record, type 129, VMS SHOW SYSTEM and SHOW MEMORY output, and several VMS lexical functions, including F$GETSYI.

1.4 - Product Prerequisites

The CA MICS VAX/VMS Analyzer runs in an environment that operates under MVS, MVS/XA, or MVS/ESA. It supports the following levels of VMS software:

VMS MONITOR versions 4.7 through 7.3.

VMS ACCOUNTING for VMS versions 5 through 7.3.

VMS ANALYZE/DISK_STRUCTURE Utility for VMS versions 5 through 7.3.

VMS SYSMAN Utility for VMS versions 5 through 7.3, specifically for the DISKQUOTA SHOW command.

You may also want to consider installing SAS/GRAPH because some of the MICF inquiries distributed with the CA MICS VAX/VMS Analyzer use its facilities.

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 2

USAGE GUIDELINES

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 2

USAGE GUIDELINES

CHAPTER 2. USAGE GUIDELINES

The CA MICS database contains VMS performance and utilization data for:

- o CPU activity
- o I/O activity
- o disk device activity
- o paging and swapping activity
- o memory usage
- o DECnet activity
- o DECdtm transaction processing
- o system initialization and status
- o LOGIN failures
- o print queues
- o lock management activity
- o file system cache
- o process and image activity
- o disk space usage and quotas

The following sections describe methods for using the VAX/VMS Analyzer's data and the interfaces between the VAX/VMS Analyzer and other CA MICS products:

1 - Data Analysis
2 - CA MICS Product Interfaces

2.1 - Data Analysis

The following identifies methods of analyzing the CA MICS VAX/VMS Analyzer's data:

Accounting and Chargeback

The VAX/VMS Analyzer provides resource usage data in a form that can be used for billing VMS system use. Usage such as consumed resources (e.g., CPU time), physical I/Os (e.g., the number of direct I/Os), and disk space occupancy (e.g., megabyte hours of storage) can be measured and charged by user defined account code levels.

Capacity

The VAX/VMS Analyzer provides utilization and workload information that is essential for determining current and future capacity requirements. System resources such as central processors, disk devices, memory, the I/O subsystem, and the paging subsystem can be examined to be sure they are not approaching site-determined capacity thresholds.

Performance

The VAX/VMS Analyzer can be used to evaluate the performance of system resources. Disk device I/O times and queuing times can be analyzed to ensure they are not causing response-time problems. Memory use and paging subsystem performance can be evaluated to ensure they are not detracting from system throughput.

Workload Characterization

A necessary task in any capacity evaluation effort is workload characterization. The VAX/VMS Analyzer can provide data to determine a system's primary workload components and the amount of resources consumed by those components. It is possible to determine, on an interval basis, average CPU utilization, number of I/Os, storage requirements, etc.

Trends

You can identify trends by comparing measurements taken in the current month with similar measurements from the previous month. Some uses of trend analysis include analyzing CPU utilization, I/O counts, and workload.

Configuration

You can use the VAX/VMS Analyzer to support utilization studies to determine if the configuration is adequate for the needs of the enterprise. If the utilization is inappropriate for the users' needs, appropriate budgetary actions can be taken.

Such analysis, performed daily or weekly, will indicate unused hardware on the system and provide the necessary information for operations to properly distribute the work over all the resources. This will minimize the under- or over-utilization of devices.

Problem Resolution

When system-related problems occur, it is not always easy to determine where and how they originated and what is impacted. The VAX/VMS Analyzer can be used to assist in determining when a problem first occurred, what workload component was responsible, and the system components affected by the problem.

Bottlenecks

The VAX/VMS Analyzer can be a valuable tool in analyzing bottlenecks to identify the causes that result in poor usage of the data center's capacity. The analysis can be used to maximize system usage and balance the distribution of the workload over the available resources.

2.2 - CA MICS Product Interfaces

Data from the CA MICS VAX/VMS Analyzer lends significant value to the use of other CA MICS products. These uses are described briefly in the following sections:

1 - Accounting and Chargeback Interface
2 - Capacity Planner Interface

2.2.1 - Accounting and Chargeback Interface

The CA MICS VAX/VMS Analyzer resource usage files contain data that can be used for billing VMS system use. This data includes CPU time, I/O counts, and disk space usage.

See the CA MICS Accounting and Chargeback Guide for information about the VAX/VMS Analyzer files and elements that are used.

Accounting and Chargeback for Disk Space Usage

You can use CA MICS Accounting and Chargeback to apply a rate per megabyte-hour against the space occupied by a file or group of files in VMS, as recorded in the DESDKU file. A comprehensive series of ISPF panels guide the accounting administrator through the process of assigning rates and other setup activities.

The result of this administrative setup is a series of SAS macros that are strategically invoked by the CA MICS VAX/VMS Analyzer during the DAILY run. No special pass of the data is required to compute and store the cost; accounting is, therefore, an optional but seamless extension to the processing involved in updating the CA MICS database.

The VAX/VMS Analyzer offers a choice of DETAIL or DAYS level accounting support for VMS disk space usage processed from the DESDKU file. DETAIL accounting allows each individual file to be priced according to the space occupied over time. DETAIL allows the customized accounting algorithms access to variables such as UIC group and member, directory name, and file name (DEXGRP, DEXMBR, DKUDIR, and DKUFILE respectively).

DAYS level accounting presents summarized observations from the DAYS level DESDKU file to the CA MICS Accounting and Chargeback algorithms. Once summarized into a single observation according to a set of sort/sequence variables, variables such as DEXGRP, DEXMBR, DKUDIR, and DKUFILE for each file are lost. The aggregate of megabyte-hours for multiple data files under a given organizational entity is presented to the accounting code, based on the disk account code assignment routine (DEXACRTD).

DAYS-level DESDKU observations presented to the accounting code are derived from the DETAIL time-span of the DESDKU file. Regardless of the selection of DETAIL or DAYS, the disk account code assignment routine is always applied to the DETAIL level of DESDKU; therefore, all DETAIL-level variables are available for deriving account codes. The account code elements are part of the set of sort/sequence variables for both DETAIL and DAYS levels.

Chapter 7 of this guide describes the parameters you must specify to properly implement accounting applications. The CA MICS Accounting and Chargeback Guide discusses various methodologies that can be used with CA MICS VAX/VMS files.

2.2.2 - Capacity Planner Interface

The CA MICS Capacity Planner can make use of the CA MICS VAX/VMS Analyzer for studies that characterize a system's workloads, track CPU utilization, analyze I/O workload, analyze paging and swapping activity, and implement numerous other analyses.

See the CA MICS Capacity Planner Guide for information on what files and elements from the CA MICS VAX/VMS Analyzer are used.

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 3

REPORTS

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 3

REPORTS

CHAPTER 3. REPORTS

The CA MICS VAX/VMS Analyzer can produce reports using the batch and interactive reporting facilities of the CA MICS IS Management Support System. Many sample reports are provided as MICF inquiries.

MICF inquiries are printer reports, color graphics, and line printer graphics that are accessed via the CA MICS Information Center Facility (MICF). MICF inquiries produce meaningful reports from the CA MICS database and provide you with the flexibility to code and save your own report formats. You can execute and easily modify any sample MICF inquiry using option 2 - Database Inquiries, from the MICF main menu.

The following topics are presented:

1 - MICF Inquiries for VAX/VMS

3.1 - MICF Inquiries for VAX/VMS

This section lists the MICF Inquiries for VAX/VMS in groupings by analysis category. The sections that follow are organized according to these category listings. Each section contains a brief description of the purpose and use of a specific CA MICS VAX/VMS Analyzer inquiry, followed by a sample output report.

It is helpful to know the inquiry time-span and output format when choosing an inquiry for analysis. MICF inquiry naming follows a standard that makes this determination possible.

Here is an example of the standard, as used for VAX/VMS Analyzer (DEX) inquiry naming:

Inquiry Name	Output format
DEXLnn	Printer Listings
DEXCnn	Color Graphics
DEXPnn	Printer Graphics

Inquiry Name	Time-span
DEXnXn	DETAIL
DEXnDn	DAYS
DEXnWn	WEEKS
DEXnMn	MONTHS
DEXnYn	YEARS

The VAX/VMS Analyzer offers the following inquiries:

VAX/VMS WORKLOAD ANALYSIS INQUIRIES

DEXLX0 - Resource Intensive Processes: (4 reports)
CPU, I/O, Image Activations, and Paging

DEXPX1 - Process Terminations Status
DEXCX1

DEXLW0 - Weekly Process Name Summary

DEXLW1 - Weekly Process Activity

DEXLM1 - Monthly Process Activity

VAX/VMS PERFORMANCE ANALYSIS INQUIRIES

DEXPX2 - Disk Response Time
DEXCX2

DEXPX3 - I/O Queue Length Average
DEXCX3

DEXPX4 - Service Performed by the System
DEXCX4

DEXPX5 - MP Synchronization Time
DEXCX5

DEXPX6 - Executive Mode Time
DEXCX6

DEXPX9 - Hard Faults to Total
DEXCX9

DEXPD7 - Available CPU Cycles
DEXCD7

DEXPD8 - Overall Page Fault Rate
DEXCD8

DEXLDH - Daily System Status - Uptime & Memory Use

DEXLDI - Daily System Summary: (2 reports)
CPU Hourly Averages, and
Average Disk Device Statistics

DEXPW2 - Weekly Average Disk Response Time
DEXCW2

DEXPW4 - Service Performed by the System - CPU Avg
DEXCW4

VAX/VMS DISK STORAGE ANALYSIS INQUIRIES

DEXLXA - Detail Directory Top Level Space

DEXLXB - Detail Directory Top Level Ranking
(2 Reports) Space Allocated & Unused

DEXLXC - Detail Directory Structure and Space

DEXLXD - Detail File Space Ranking - Allocated

DEXLXE - Detail Disk Quota Statistics

DEXLDA - Daily Directory Top Level Usage

DEXLDB - Daily Directory Top Level Ranking:
(2 Reports) Megabyte Hours Allocated & Unused

DEXPWF - Weekly Disk Device Free Space
DEXCWF

DEXPWG - Weekly Disk Device Space Utilization
DEXCWG

DEXLMA - Monthly Disk Usage Summary

DEXLMB - Monthly Disk Usage Ranking:
(2 Reports) Megabyte Hours Allocated & Unused

The following sections describe the individual inquiries:

1 - VAX/VMS Workload Analysis Inquiries
2 - VAX/VMS Performance Analysis Inquiries
3 - VAX/VMS Disk Storage Analysis Inquiries

3.1.1 - VAX/VMS Workload Analysis Inquiries

This section lists the MICF Inquiries for VAX/VMS workload analysis. A brief description is given for each CA MICS VAX/VMS Analyzer inquiry in this category. Sample report output follows this section.

DEXLX0 - Resource Intensive Processes: (4 reports)
CPU, I/O, Image Activations, and Paging

Sample: Fig 3-101 and 3-102.

This inquiry shows you in a single report who is using the largest amounts of key system resources in a single process. The report shows the top 25 resource users for each of the following measures:

- o CPU Time
- o Direct I/Os
- o Image Activations
- o Page File Peak usage

A separate report page is printed in ranked list format for each of the measures, with identification of the time and the user (process) consuming the resource. You can modify this inquiry to include other measurements of interest or expand the size of the ranking list you want to view.

DEXPX1 - Process Terminations Status
DEXCX1

Sample: Fig 3-103.

This inquiry is useful in determining the size and composition of the workload running on each VMS system, based on VMS Accounting measurements. The report shows the overall distribution of process terminations by hour of day. Each hourly measure is broken down by type of termination (for example, success, warning, error) based on the final status of the last image run by the process. This inquiry can be easily customized to provide other breakdowns, such as by process type, account codes, or user names.

DEXLW0 - Weekly Process Name Summary

Sample: Fig 3-104.

This inquiry is useful in determining the overall size and composition of the workload running on each VMS system, based on VMS Monitor measurements. The report lists process activity by week and time zone, for each process name prefix. This prefix is defined as the first part of process name (typically a user name), up to 8 characters, prior to special characters such as underscore. This grouping of process names gives a compact summary of process activity while retaining enough information to help determine the type of workload represented.

The report shows the summarized values for each of the following measures:

- o CPU Time
- o Direct I/Os
- o Page Faults
- o Total Recording Interval Time

You can modify this inquiry to include other MONITOR PROCESS measurements of interest.

DEXLW1 - Weekly Process Activity

Sample: Fig 3-105.

This inquiry is useful in determining the size and composition of the workload running on each VMS system, based on a combination of VMS Accounting and VMS Monitor measurements. The report shows the distribution of process activity by zone summed over an entire week. Each measure is broken down by hour of day, process type (for example, batch, interactive, detached), and UIC group, based on the combination of Monitor and Accounting PROCESS records from the DESPRX file.

The report shows the summarized values for each of the following measures:

- o CPU Time
- o Direct I/Os
- o Buffered I/Os
- o Page Faults
- o Maximum Global and Process Pages

You can modify this inquiry to include other MONITOR

PROCESS measurements of interest, with other breakdowns such as by user name.

DEXLM1 - Monthly Process Activity

Sample: Fig 3-106.

This inquiry is useful in determining the size and composition of the workload running on each VMS system, based on a combination of VMS Accounting and VMS Monitor measurements. The report shows the distribution of process activity summed over an entire month. Each measure is broken down by zone, process type (for example, batch, interactive, detached), and UIC group, based on the combination of Monitor and Accounting PROCESS records from the DESPRX file.

The report shows the summarized values for each of the following measures:

- CPU time
- Direct I/Os
- Buffered I/Os
- Page Faults
- Maximum Global and Process Pages

You can modify this inquiry to include other MONITOR PROCESS measurements of interest, with other breakdowns such as by user name.

CPU Time - Top 25 Processes

INQUIRY: DEXLX0 RUN DATE: 24JUL08

System Identifier=VMS2

PRCCPUTM	ENDTS	DEXUSER	DEXGRP	DEXMBR
0:02:55.54	11JUL08:00:04:45.55	JQINJQ	200	2
0:01:39.68	11JUL08:15:56:15.31	DD9RV54	1	1
0:01:14.44	11JUL08:11:56:47.20	JQINJQ	200	2
0:01:00.03	11JUL08:13:25:46.58	JQINJQ	200	2
0:00:56.53	11JUL08:15:56:14.60	DD9RV54	1	1
0:00:45.06	11JUL08:14:33:04.01	DD9RV54	1	1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Direct I/Os - Top 25 Processes

INQUIRY: DEXLX0 RUN DATE: 24JUL08

System Identifier=VMS2

PRCDIRIO	ENDTS	DEXUSER	DEXGRP	DEXMBR
2872	11JUL08:00:04:45.55	JQINJQ	200	2
1922	11JUL08:00:08:34.64	JQINJQ	200	2
1758	11JUL08:13:25:46.58	JQINJQ	200	2
468	11JUL08:11:56:47.20	JQINJQ	200	2
356	11JUL08:15:56:14.60	DD9RV54	1	1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Figure 3-101. Resource Intensive Processes - CPU & I/O

Image Activations - Top 25 Processes

INQUIRY: DEXLX0 RUN DATE: 24JUL08

System Identifier=VMS2

PRCIMAGE	ENDTS	DEXUSER	DEXGRP	DEXMBR
56	11JUL08:15:56:14.60	DD9RV54	1	1
54	11JUL08:00:04:45.55	JQINJQ	200	2
33	11JUL08:11:56:47.20	JQINJQ	200	2
11	11JUL08:13:25:46.58	JQINJQ	200	2
5	11JUL08:16:50:54.08	DD9GW20	200	73
5	11JUL08:17:54:31.66	DD9GW20	200	73
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Page File Peak Usage - Top 25 Processes

INQUIRY: DEXLX0 RUN DATE: 24JUL08

System Identifier=VMS2

PRCPAGFL	ENDTS	DEXUSER	DEXGRP	DEXMBR
11555	11JUL08:18:16:19.87	DD9GW20	200	73
10402	11JUL08:00:04:45.55	JQINJQ	200	2
10139	11JUL08:16:50:54.08	DD9GW20	200	73
10034	11JUL08:17:54:31.66	DD9GW20	200	73
10034	11JUL08:14:36:09.03	MA9KM38	300	10
7028	11JUL08:14:47:49.80	DD9RV54	1	1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Figure 3-102. Resource Intensive Processes - Images & Paging

```
INQUIRY:  DEXPX1                                                
```

Weekly Process Name Summary

INQUIRY: DEXLW0 RUN DATE: 1AUG08

SYSID	Proc Name Prefix	Week	Time Zone	CPU Time	Direct I/Os	Buffered I/Os	Page Faults	Interval Time	Week Date
VMS2	BATCH	18	1	0:00:42.76	2,066	126	0	27:01:44.25	02MAY2008
		18	2	0:01:11.06	3,392	206	958	44:47:37.82	01MAY2008
		18	3	0:00:38.83	1,816	110	322	24:01:24.71	05MAY2008
		19	1	0:01:08.12	3,297	201	323	43:17:44.47	06MAY2008
		19	2	0:01:56.37	5,644	342	1,713	74:19:37.87	06MAY2008
		19	3	0:01:00.96	2,879	175	643	38:02:13.85	05MAY2008
		24	1	0:00:49.51	2,325	142	646	30:03:07.78	12JUN2008
		24	2	0:01:23.99	3,979	243	998	52:04:14.31	11JUN2008
		24	3	0:00:37.47	1,803	110	316	23:46:21.48	16JUN2008
		25	1	0:03:27.44	5,386	1,863	9,110	45:03:07.91	17JUN2008
		25	2	0:01:59.77	5,713	347	1,633	75:05:00.75	17JUN2008
		25	3	0:01:16.69	3,645	222	640	48:02:50.14	16JUN2008
		26	1	0:01:11.91	3,468	211	0	45:03:17.70	24JUN2008
		26	2	0:04:17.78	7,937	1,979	9,059	74:50:04.98	24JUN2008
		26	3	0:01:14.93	3,644	222	642	48:02:50.85	23JUN2008
		27	2	0:00:00.39	18	1	0	0:15:00.85	01JUL2008
		27	3	0:00:38.13	1,822	111	333	24:01:25.84	30JUN2008
----	--------			---	---	---	---	---	
VMS2	BATCH			0:23:36.11	58,834	6,611	27,336	717:47:45.56	
VMS2	BLUNE	24	1	0:00:00.44	7	20	185	30:03:07.78	17JUN2008
		24	2	0:00:00.00	0	0	0	52:04:14.31	17JUN2008
		24	3	0:00:00.00	0	0	0	23:46:21.48	16JUN2008
		25	1	0:00:00.00	0	0	0	45:03:07.91	20JUN2008
		25	2	0:00:00.00	0	0	0	75:05:00.75	20JUN2008
		25	3	0:00:00.00	0	0	0	48:02:50.14	23JUN2008
		26	1	0:00:00.00	0	0	0	45:03:17.70	25JUN2008
		26	2	0:00:01.98	11	103	136	74:50:04.98	25JUN2008
		26	3	0:00:00.00	0	0	0	48:02:50.85	27JUN2008
		27	2	0:00:00.00	0	0	0	0:15:00.85	01JUL2008
		27	3	0:00:00.00	0	0	0	24:01:25.84	30JUN2008
----	--------			---	---	---	---	---	
VMS2	BLUNE			0:00:02.42	18	123	321	466:17:22.59	
VMS2	CD9JE15	19	1	0:00:16.68	183	506	2,467	1:30:06.69	10MAY2008
		19	2	0:00:25.49	289	1,737	4,155	2:00:08.10	09MAY2008
		24	1	0:00:10.26	73	623	1,140	2:30:52.54	12JUN2008
		26	1	0:00:05.77	68	235	787	0:15:01.03	27JUN2008
----	--------			---	---	---	---	---	
VMS2	CD9JE15			0:00:58.20	613	3,101	8,549	6:16:08.36	

Figure 3-104. Weekly Process Name Summary

Weekly Process Activity

INQUIRY: DEXLW1 RUN DATE: 1AUG08

SYSID = VMS2 Year = 08 Week = 25 Time Zone = 1

Hour of Day	Process Type	UIC Group	CPU Time	Direct I/O	Buffered I/O	Page Faults	Max Global Pages	Max Proc Pages
8		.	0:01:39.56	663	3,542	18,332	288	604
8	BATCH	200	0:00:07.83	367	20	0	212	760
8	DETACHED	1	0:00:05.97	0	0	0	404	487
8	INTERACT	1	0:00:00.00	0	0	0	72	144
8	INTERACT	200	0:00:00.00	0	0	0	78	245
			0:01:53.36	1,030	3,562	18,332		
9		.	0:01:42.70	664	3,664	18,375	288	604
9	BATCH	200	0:00:08.02	391	25	0	212	760
9	DETACHED	1	0:00:05.89	0	0	0	404	487
9	INTERACT	1	0:00:00.00	0	0	0	72	144
9	INTERACT	200	0:03:02.10	1,929	4,969	21,379	1,065	929
			0:04:58.71	2,984	8,658	39,754		
10		.	0:01:50.87	761	3,977	18,985	288	604
10	BATCH	200	0:02:12.43	2,170	1,507	7,273	212	1,143
10	DETACHED	1	0:00:17.75	52	630	0	404	487
10	INTERACT	1	0:00:00.00	0	0	0	72	144
10	INTERACT	200	0:02:29.55	1,478	6,021	25,376	161	608
			0:06:50.60	4,461	12,135	51,634		
11		.	0:01:55.86	1,217	3,829	18,324	288	604
11	BATCH	200	0:00:19.45	527	191	1,837	212	760
11	DETACHED	1	0:00:06.77	7	54	1	404	487
11	INTERACT	1	0:00:09.03	68	568	632	75	194
11	INTERACT	200	0:03:11.36	2,515	6,933	37,450	1,138	935
			0:05:42.47	4,334	11,575	58,244		
12		.	0:01:42.98	697	3,740	18,489	288	604
12	BATCH	200	0:00:07.89	394	25	0	212	760
12	DETACHED	1	0:00:13.48	30	390	0	404	487
12	INTERACT	1	0:00:00.00	0	0	0	72	144
12	INTERACT	200	0:01:10.19	605	1,685	6,140	125	259
			0:03:14.54	1,726	5,840	24,629		

Figure 3-105. Weekly Process Activity

Monthly Process Activity

INQUIRY: DEXLM1 RUN DATE: 1AUG08

SYSID = VMS2 Year = 08 Month = 6

Time Zone	Process Type	UIC Group	CPU Time	Direct I/O	Buffered I/O	Page Faults	Max Global Pages	Max Proc Pages
1		.	0:43:34.46	18,109	84,287	383,115	397	604
1	BATCH	200	0:05:28.86	11,179	2,216	9,756	221	1,143
1	DETACHED	1	3:19:03.45	1,701	715,524	27,880	454	599
1	INTERACT	1	2:36:26.30	4,543	430,480	1,145,359	1,232	963
1	INTERACT	10	0:00:30.18	321	1,057	5,343	1,285	1,139
1	INTERACT	100	0:00:15.29	86	593	1,812	78	171
1	INTERACT	200	0:47:42.07	27,030	97,798	285,616	1,514	1,144
1	INTERACT	201	0:00:00.99	6	32	105	109	203
			7:33:01.60	62,975	1,331,987	1,858,986		
2		.	1:07:33.37	27,671	132,737	639,967	397	604
2	BATCH	200	0:07:41.54	17,629	2,569	11,690	221	1,151
2	DETACHED	1	1:35:25.74	1,304	345,351	10,822	454	599
2	INTERACT	1	0:58:40.68	250	164,727	438,472	127	226
2	INTERACT	100	0:00:22.81	581	716	870	76	139
2	INTERACT	200	0:38:18.06	25,250	82,759	133,033	1,367	1,105
2	INTERACT	201	0:00:09.75	90	274	1,220	112	200
			4:28:11.95	72,775	729,133	1,236,074		
3		.	0:49:37.98	21,104	104,195	490,049	395	698
3	BATCH	200	0:03:09.47	9,110	555	1,598	218	758
3	DETACHED	1	0:03:12.93	319	4,085	11	404	489
3	INTERACT	1	0:00:00.00	0	0	0	72	144
3	INTERACT	200	0:00:00.40	0	8	123	24	139
			0:56:00.78	30,533	108,843	491,781		
			15:34:46.14	264,515	2,379,704	4,066,683		

Figure 3-106. Monthly Process Activity

3.1.2 - VAX/VMS Performance Analysis Inquiries

This section lists the MICF inquiries for VAX/VMS performance analysis. A brief description is given for each CA MICS VAX/VMS Analyzer inquiry in this category. Sample report output follows this section.

DEXPX2 - Disk Response Time
DEXCX2

Sample: Fig 3-201.

This inquiry is useful for easily spotting unusual behavior or performance problems with a disk unit or controller. It is also useful for comparing the relative performance of your disks. The report shows a detailed view of average (sustained) response time for each disk unit on each controller on the system. The plots are grouped together by controller, with data points shown to as fine a level of detail as your MONITOR collection interval.

DEXPX3 - I/O Queue Length Average
DEXCX3

Sample: Fig 3-202.

This inquiry helps you analyze disk response time by showing which measurements merit further analysis. The report shows a detailed view of average (sustained) I/O queue length for each disk unit on each controller on the system. The plots are grouped together by controller, with data points shown to as fine a level of detail as your MONITOR collection interval. The graph area is divided into 3 parts: queue length less than .2, queue length from .2 to .5, and queue length greater than .5. These correspond to light, moderate, and heavily busy categories for disk activity, and thus indicate the potential impact any improvements will have on overall system performance.

DEXPX4 - Service Performed by the System
DEXCX4

Sample: Fig 3-203.

This inquiry lets you determine whether the system itself is consuming excessive resources, and thus can help you identify a source of additional CPU cycles. The report shows a detailed view by CPU of interrupt stack and kernel mode time as a percentage of total CPU time, depicted cumulatively on the chart. This view lets you see each individual measurement (DEXCX4 only) plus the cumulative value representing service performed by the system (actually two types of kernel mode time). The graph area is divided into two parts, to indicate the general rule that the combined measurement should not exceed 40%.

DEXPX5 - MP Synchronization Time
DEXCX5

Sample: Fig 3-204.

This inquiry lets you identify potentially high levels of paging, I/O, or locking activity, which leads to further investigation and improvement for the affected resource. The report shows a detailed view by CPU of MP synchronization time as a percentage of total CPU time, with data points shown to as fine a level of detail as your MONITOR collection interval. This component of CPU resource consumption is a measure of contention for spinlocks (synchronization mechanisms) in a multiprocessing system. The graph area is divided into two parts, to indicate the general rule that above 8% indicates moderate to heavy activity.

DEXPX6 - Executive Mode Time
DEXCX6

Sample: Fig 3-205.

This inquiry lets you identify potentially high levels of RMS or other database activity, which in turn affect the CPU resource. The report shows a detailed view by CPU of executive mode time as a percentage of total CPU time, with data points shown to as fine a level of detail as your MONITOR collection interval. This component of CPU resource consumption will vary based on your use of these facilities, including such factors as file designs and access characteristics.

DEXPX9 - Hard Faults to Total
DEXCX9

Sample: Fig 3-207.

This inquiry helps you analyze use of the secondary page cache (free list and modified list). Paging problems typically occur when this system-wide cache is too small. The report shows a detailed view of hard faults (page read I/Os) as a percentage of overall page faults on the system. The plots are produced by system, with data points shown to as fine a level of detail as your MONITOR collection interval. This component of memory usage should be kept as low as possible. The graph area is divided into two parts, to indicate the general rule that a hard fault to total ratio above 10% indicates inefficient use of secondary cache.

DEXPD7 - Available CPU Cycles
DEXCD7

Sample: Fig 3-208.

This inquiry shows you the relative size of your workload by hour of day and indicates potential sources of extra processing time using your current configuration. The report shows the overall availability of processing time (idle time) by hour of day, with each hourly measure broken down by CPU identifier. You can view this report for as many days as you want to input to the inquiry.

DEXPD8 - Overall Page Fault Rate
DEXCD8

Sample: Fig 3-209.

This inquiry helps you analyze memory usage by establishing what constitutes normal acceptable page fault behavior on each of your systems, thus also letting you spot potential memory-related problems. The report shows hourly average page fault rates for hard and soft faults (excluding system faults). The chart area is divided into four parts using reference lines at rates of 30, 60, and 100 per second. These correspond to guidelines for acceptable ranges for various classes of processors. For example, paging above 100 per second on an 11/780 is a warning that attention may be needed in the area of memory resources.

DEXLDH - Daily System Status - Uptime & Memory Use

Sample: Fig 3-210.

This inquiry displays several of the generally useful measures carried in the system status file, reporting one observation for each day. System uptime lets you see at a glance how long your system has maintained continuous operation. The report also shows key memory usage initial settings and maximum actual values reached. Other measures are available to this report such as total physical memory (useful for memory utilization calculations), and other elements from DESSYU file.

DEXLDI - Daily System Summary: (2 reports)
CPU Hourly Averages, and
Average Disk Device Statistics

Sample: Fig 3-211 and 3-212.

This inquiry consolidates key system performance measures into a 2-page report for each system. The first page covers CPU related statistics for each hour of the day. The second page covers disk device performance and overall storage statistics, for each device.

CPU measures include the following:

- o Average CPU Percent Busy
- o System Percent of CPU
- o User Percent of CPU
- o Page Faulting and Inswap rates
- o Direct and Buffered I/O rates

Disk device measures include the following:

- o Device Type
- o Blocks Total, Blocks Free, and Percent Usage
- o Average Response Time and I/O Rate
- o Volume Name and Volume Set Name
- o Last (most recent) Error Count

This report gives you a comprehensive look at system performance by hour over as many days as you choose at report run time.

DEXPW2 - Weekly Average Disk Response Time
DEXCW2

Sample: Fig 3-213.

This inquiry is useful for analyzing disk response time long term behavior and trends, and for spotting unusual behavior or performance problems with a disk device. It is also useful for comparing the relative performance of your disks during different levels of system activity. The report shows a summary view of average (sustained) response time for each disk device on the system. Each bar on the chart represents a time zone summary for an entire week.

DEXPW4 - Service Performed by the System - CPU Avg
DEXCW4

Sample: Fig 3-214.

This inquiry helps you to determine whether the system itself is consuming excessive resources, and thus can help you identify a source of additional CPU cycles. The report shows an average hourly view of interrupt stack and kernel mode time as a percentage of total CPU time, averaged across CPUIDs, depicted cumulatively on the chart. This view lets you see each individual measurement (DEXCW4 only) plus the cumulative value representing service performed by the system (actually two types of kernel mode time). The graph area is divided into two parts, to indicate the general rule that the combined measurement should not exceed 40%.

```
+-----------------------------------------------------------------------------------------------------------------------------------------+
|                                                          Disk Response Time                                                             |
| INQUIRY:  DEXPX2                                                                                                          RUN DATE:  24JUL08|
|                                                SYSID=VMS1 Node=HSC005 Controller=DUA                                                    |
|                                                                                                                                         |
|                                          PLOT OF DSKAVRES*ENDTS    SYMBOL IS VALUE OF DSKUNIT                                          |
|  0:00:00.550 +                                                                                                                          |
|              |                                                                                                                          |
|              |        1                                                                                                                 |
|              |                                                                                                                          |
|  0:00:00.500 +                                                                                                  2                       |
|              |                                                                                                                          |
|              |                                                                                                                          |
|              |                                                                                                                          |
|  0:00:00.450 +                                                                                                                          |
|D             |                                                                                                                          |
|i             |                                                                                                                          |
|s             |                                                                                                                          |
|k 0:00:00.400 +                                                                                                                          |
|              |                                                                                                                          |
|A             |                                                                                                                          |
|v             |                                                                                                                          |
|e 0:00:00.350 +                                                                                                                          |
|r             |                                                                                                                  2       |
|a             |                                                                                                                          |
|g             |                                                                                                                          |
|e 0:00:00.300 +                                                                                                                          |
|              |                                                                                                                          |
|R             |                                                                                                                          |
|e             |                                                                                                                          |
|s 0:00:00.250 +                                                                                                                          |
|p             |                                                                                                                          |
|o             |                                     1                                                                                   |
|n             |                                                                                                                          |
|s 0:00:00.200 +                                                                                                                          |
|e             |                                                                                                                          |
|              |                                                                                                                          |
|T             |                                                                                                                          |
|i 0:00:00.150 +           1                                                                                                              |
|m             |      1                                                                                                                   |
|e             |         2                                                                                                                |
|              |     1              2     1  2  22 22 2  2 22                                                  2                          |
|  0:00:00.100 +        1        22  2122 222 2   2     2    2                                                                           |
|              |                                     2      1                                                                    1       |
|              |                                                                                2                                         |
|              |    1 1  1    1 2                                                                21              1            1           |
|  0:00:00.050 +                    11                        11   1       2   22 2 2 122 2  21             2       2 2   1              |
|              |   11       1    11           1    1                1  1  2   111   11  112      11  1111  1112 1  11 122               |
|              |     1    1    11 1 1  11   11  111 1 1    1   1  111  11111 111 1 11  1 11111 11111 11111 11  1 111   1 12  111 1      |
|              |                         1                                     11                  11                        1 1          |
|  0:00:00.000 +         12 1          1     11         11       1  11                                          2     1   1             |
|             -+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+|
|       16JUL08:19:40:00    16JUL08:23:50:00    17JUL08:04:00:00    17JUL08:08:10:00    17JUL08:12:20:00    17JUL08:16:30:00          |
|                                                                   End Time Stamp                                                        |
|NOTE:    167 OBS HAD MISSING VALUES      29 OBS HIDDEN                                                                                   |
|                                                                                                                                         |
|                                                Figure 3-201.   Disk Response Time                                                       |
+-----------------------------------------------------------------------------------------------------------------------------------------+
```

```
+-----------------------------------------------------------------------------------------------------------------------------------+
|                                                   I/O Queue Length Average                                                        |
| INQUIRY:  DEXPX3                                                                                                   RUN DATE:  24JUL08|
|                                                SYSID=VMS1 Node=HSC005 Controller=DUA                                              |
|                                                                                                                                   |
|                                        PLOT OF AVQUELEN*ENDTS    SYMBOL IS VALUE OF DSKUNIT                                       |
|        |                                                                                                                          |
|  1.40 +                                                                                                                           |
|        |                                                                                                                          |
|        |                                                1                                                                         |
|  1.30 +                                                                                                                           |
|        |                                                                                                                          |
|        |                                                                                                                          |
|  1.20 +                                                                                                                           |
|        |                                                                                                                          |
|        |                                                                                                                          |
|  1.10 +                                                                                                                           |
|A       |                                                                                                                          |
|v       |                                                                                                                          |
|e 1.00 +                                                                                                                           |
|r       |                                                                                                                          |
|a       |                                                                                                                          |
|g 0.90 +                                                                                                                           |
|e       |                                                                                                                          |
|        |                                                                                                                          |
|Q 0.80 +                                                                                                                           |
|u       |                                                                                                                          |
|e       |                                                                                                                          |
|u 0.70 +                                                                                                                           |
|e       |                                                                                                                          |
|        |                                                                                                                          |
|L 0.60 +                                                                                                                           |
|e       |                                                                                                                          |
|n       |                                                                                                                          |
|g 0.50 +-------------------------------------------------------------------------------------------------------------------------|
|t       |                                                                                                                          |
|h       |         1                                                                                                                |
|  0.40 +                                   2                                                                                       |
|        |                     2   2   22  22 22    2 222                                                                           |
|        |                  2  222  22  2   2    22                                                                                  |
|  0.30 +                 2                                                                                                         |
|        |                                                                                  2 1                                      |
|        |                  1 1                                                                                                      |
|  0.20 +-------------------------------------------------------------------------------------------------------------------------|
|        |                     1                                          2             1               1         2                |
|        |                                  1        1    1           2       1  1                                   1            |
|  0.10 +                 1             1            21                          1              2   1 1   2                        |
|        |                                                                  1  1  2 22              1           22                 |
|        |      1    1    1  1   11    1 1  1 1  1    1      1 1    1    1111 1111  1111  1 1 1  1111 11111 121     1    1    1   |
|  0.00 +    1111 11111 11113 11111 11111 11111 1111 11111 11111 11111 11123 22131 33131 11111 11112 33212 21211 11111 11111 11    |
|       --+------------------+------------------+------------------+------------------+------------------+------------------+-----|
| 16JUL08:19:40:00    16JUL08:23:50:00    17JUL08:04:00:00    17JUL08:08:10:00    17JUL08:12:20:00    17JUL08:16:30:00            |
|                                                                 End Time Stamp                                                    |
|NOTE:     181 OBS HIDDEN                                                                                                           |
|                                                                                                                                   |
|                                                                                                                                   |
|                                          Figure 3-202.   I/O Queue Length Average                                                 |
+-----------------------------------------------------------------------------------------------------------------------------------+
```

Service Performed by the System

INQUIRY: DEXPX4 RUN DATE: 24JUL08

SYSID=VMS1 CPUID=3

PLOT OF SRVPCSYS*ENDTS SYMBOL USED IS *

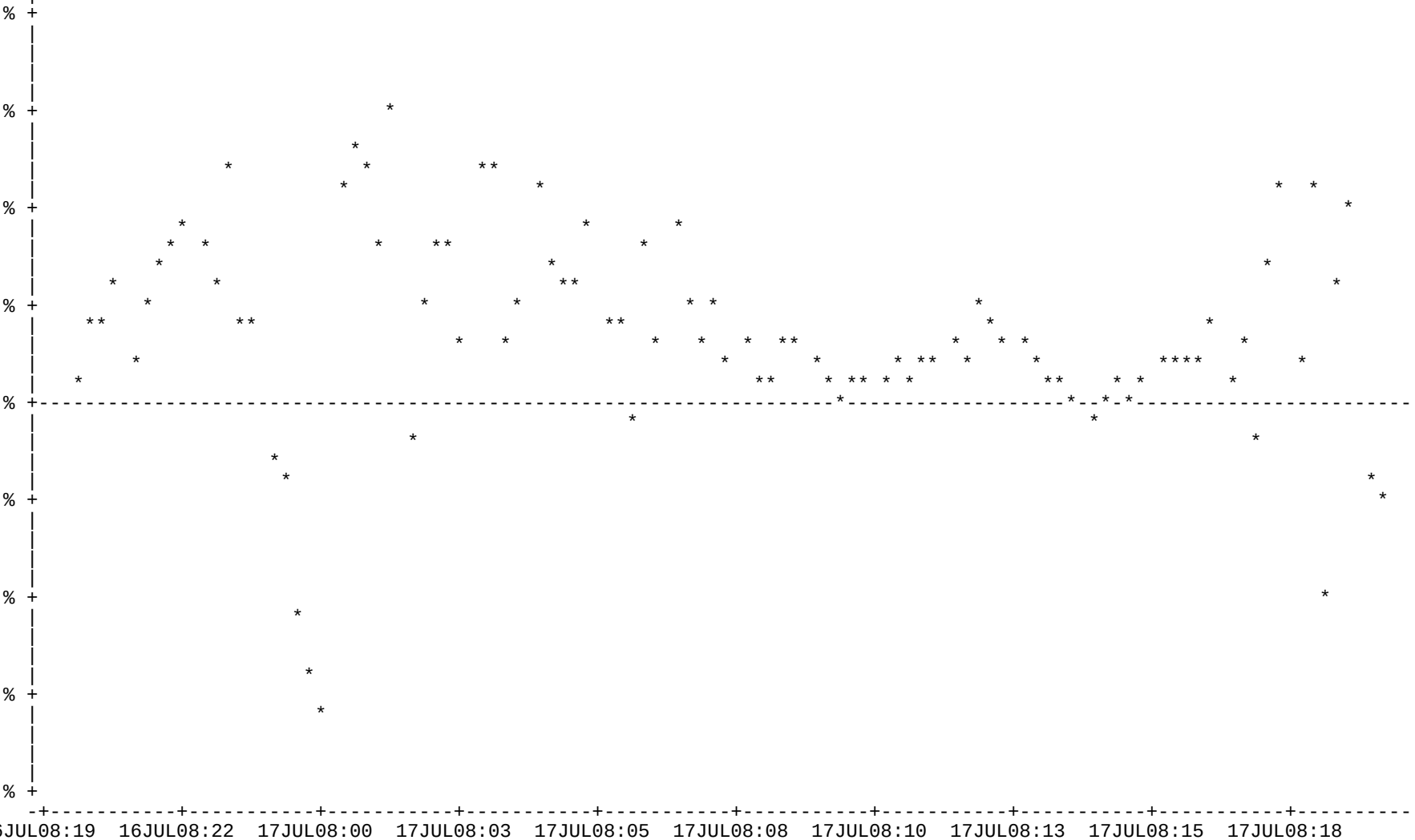

Figure 3-203. Service Performed by the System

```
                                                  MP Synchronization Time
 INQUIRY:  DEXPX5                                                                                                  RUN DATE:  24JUL08
                                                         SYSID=VMS1

                                       PLOT OF MPSPCCPU*ENDTS    SYMBOL IS VALUE OF MODCPU

            |
  20.00 % +
            |
            |
            |                 2
            |
  17.50 % +                 3
            |
M           |
P           |
            |
S 15.00 % +
y           |
n           |
c           |
            |
T 12.50 % +
i           |
m           |
e           |
            |                                 1
P 10.00 % +
c           |                                 3
t           |                                 2                                   2                 2    2  2
            |                                                               22     2  22           2  3  223  32
o           |----------------1-----------------------------------------22-32-----3-332-2----------2--2-3-2----------------------
f  7.50 % +                 2                                          33 3 2 23     23             33  3  3 2
            |   2           31                                                 2      33           23          23
C           |   3                                                       2       33              2 2 3
P           |                                                           3                  2  23
U           |      2   3                                                2
   5.00 % +     2   222                                                 3                     22
            |     3  2                                                  2                                          2          3
            |    2      3  3                                      2  23                                            3
            |    3   3     23                                        2                                              2     2  22
            |            2  2                                   333       11                                          2    3   3 2
   2.50 % +   1                                          2 2   22      1   111     1 11           1 1        1  1    223       32
            |                3 22    33  2 2 22  2    3  2   2       1        1111    111 1   1 1 1 1 11111  11     3 2  23    3
            |    1 1           2 33    2222   2     222 22    331 1  11                      111                  1  1      2
            |     1  111    1                1      133       21    1                                              1 1     1
            |           11 1   1 11212 111 1  111  1  1 1 111    1                                                   11 11 11 11
   0.00 % +                      1 1    1          1   1    1
            -+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
   16JUL08:19:40:00    16JUL08:23:50:00    17JUL08:04:00:00    17JUL08:08:10:00    17JUL08:12:20:00    17JUL08:16:30:00
                                                            End Time Stamp
NOTE:      35 OBS HIDDEN
```

Figure 3-204. MP Synchronization Time

```
+-----------------------------------------------------------------------------------------------------------------------------------------+
|                                                   Executive Mode Time                                                                   |
| INQUIRY:  DEXPX6                                                                                                          RUN DATE:  24JUL08|
|                                                        SYSID=VMS1                                                                       |
|                                     PLOT OF EXEPCCPU*ENDTS    SYMBOL IS VALUE OF MODCPU                                               |
|  27.50 % +                                                                                                                              |
|          |                                                                                                                              |
|          |                                                                                                                              |
|          |               3                                                                                                              |
|  25.00 % +              2                                                                                                               |
|          |              32                                                                                                              |
|          |                                                                                                                              |
|          |                                                                                                                              |
|  22.50 % +                                                                                                                              |
|          |                                                                                                                           2  |
|E         |                                                                                                                        3     |
|x         |                                                                                                                              |
|e 20.00 % +                                                   2                                                                  2  3  |
|c         |                                                2      3                                                                    |
|u         |                                                                                                                              |
|t         |                                   3                                                                                       2  |
|i 17.50 % +            2              2                                                                                                |
|v         |            3                                     2                                                                          |
|e         |                        3     2                                                                                               |
|          |                           2                      3                                                                          |
|M 15.00 % +                          2         2  3                                                                                    |
|o         |                              3                                                                                               |
|d         |                            3   2       2   3 2 32                                                                            |
|e         |                 2          3 2     23      2        2   22                                                          3       |
|  12.50 % +                        2         2        2      3  2 32                                   2    2         22              |
|P         |                     3              3   23               3  2       2                                23                   |
|c         |                     22                     3    2         3 22  2                                  3   22 3   3          |
|t         |   2            3                        3    2                322  32222 2 22        232    2          33 22              |
|  10.00 % +                     3        3                              33        23 2 22 32 32 22  323      2     3                |
|o         |                                  2  3                                  3   22          2  22  3                            |
|f         |           2                        2                                                2             2                          |
|          |   1       3                                                                                                                |
|C  7.50 % +        2                                                1 1   111                                                         |
|P         |                                                          1  11    1                          1   11                        |
|U         |    2   3 2       2                                         1 1    1          1    1                                       |
|          |     22                                        1 1  1           1 1  111 1       11 1                                        |
|   5.00 % +    333  23       3                                                      11 11     1 1 11  11                    2         |
|          |    1   13                                                                1                    1 11                          |
|          |     11        1                1   1                                                            1     1   32              |
|          |                   1 22                            1                                                 1        1            |
|   2.50 % +         11   1      1332                     1     1                                                      1   1           |
|          |           11    1    111 1   1    1   1     1    1                                                    1   1               |
|          |                131        11    1    1       1                                                     1  1                 |
|          |                             1    1     11 11    1                                                                          |
|   0.00 % +                                                                                                                              |
|         -+------------------+------------------+------------------+------------------+------------------+------------------+----------|
|   16JUL08:19:40:00    16JUL08:23:50:00    17JUL08:04:00:00    17JUL08:08:10:00    17JUL08:12:20:00    17JUL08:16:30:00              |
|                                                                 End Time Stamp                                                          |
|NOTE:      35 OBS HIDDEN                                                                                                                 |
|                                                                                                                                         |
|                                                                                                                                         |
|                                          Figure 3-205.   Executive Mode Time                                                            |
+-----------------------------------------------------------------------------------------------------------------------------------------+
```

Hard Faults to Total

INQUIRY: DEXPX9 RUN DATE: 24JUL08

SYSID=VMS1

PLOT OF HRDPCTOT*ENDTS SYMBOL USED IS *

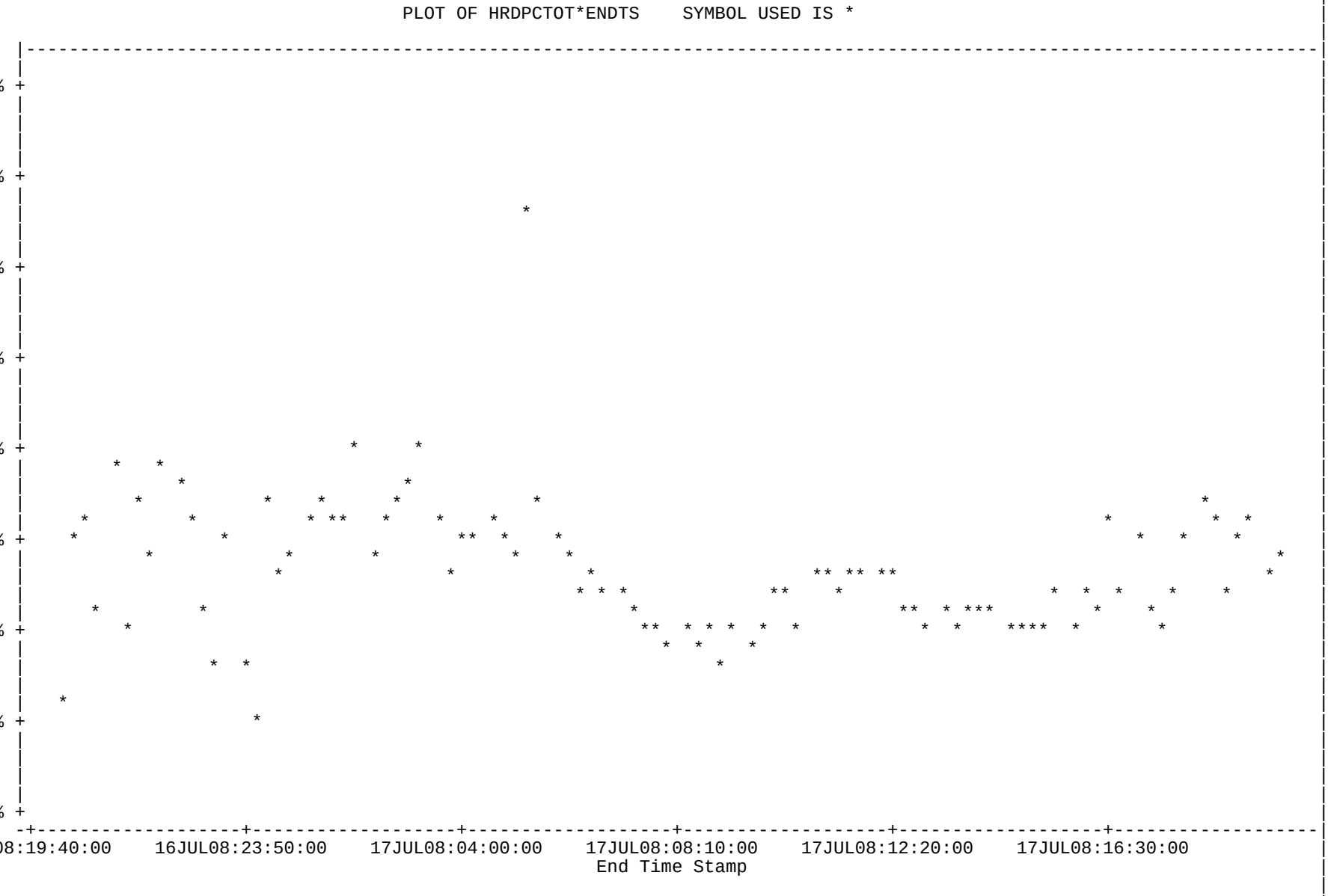

Figure 3-207. Hard Faults to Total

```
+-----------------------------------------------------------------------------------------------------------------------------------------+
|                                                   Available CPU Cycles                                                                  |
| INQUIRY:  DEXPD7                                                                                                   RUN DATE:  DDMMMYY |
|                                                 SYSID=VMS1 Month=7 Day=17                                                               |
|                                                     BAR CHART OF SUMS                                                                   |
|MODIDLTM SUM                                                                                                                             |
|           |                                                                                                                                 |
|2:46:40.00 +      3333  3333                                                                                                             |
|           |      3333  3333  3333  3333                                                                                                     |
|           |      3333  3333  3333  3333                                                                                                     |
|           |      3333  3333  3333  3333  3333                                                                                               |
|2:30:00.00 +      3333  3333  3333  3333  3333                                                                                               |
|           |      3333  3333  3333  3333  3333                                                                                       3333    |
|           |      3333  3333  3333  3333  3333  3333                                                                                  3333    |
|           |      3333  3333  3333  3333  3333  3333                                                                                  3333    |
|2:13:20.00 +      3333  3333  3333  3333  3333  3333                                                                                  3333    |
|           |      3333  3333  3333  3333  3333  3333                                                                          3333    3333    |
|           |      3333  3333  3333  3333  3333  3333                                                                          3333    3333    |
|           |      3333  3333  3333  3333  3333  3333                                                                          3333    3333    |
|1:56:40.00 +      3333  3333  3333  3333  3333  3333                                                                          3333    3333    3333|
|           |3333  2222  2222  3333  3333  3333  3333                                                                          3333    3333    3333|
|           |3333  2222  2222  2222  2222  3333  3333  3333                                                                    3333    3333    3333|
|           |3333  2222  2222  2222  2222  2222  3333  3333                                                                    3333    3333    3333|
|1:40:00.00 +3333  2222  2222  2222  2222  2222  3333  3333                                                                    3333    2222    3333|
|           |3333  2222  2222  2222  2222  2222  2222  3333                                                                    3333    2222    3333|
|           |3333  2222  2222  2222  2222  2222  2222  3333                                        3333                        3333    2222    3333|
|           |3333  2222  2222  2222  2222  2222  2222  3333                                        3333                        2222    2222    3333|
|1:23:20.00 +3333  2222  2222  2222  2222  2222  2222  3333                                        3333                        2222    2222    3333|
|           |2222  2222  2222  2222  2222  2222  2222  3333                                        3333                        2222    2222    2222|
|           |2222  2222  2222  2222  2222  2222  2222  2222                                        3333                        2222    2222    2222|
|           |2222  2222  2222  2222  2222  2222  2222  2222                                        3333                        2222    2222    2222|
|1:06:40.00 +2222  2222  2222  2222  2222  2222  2222  2222                                        3333                        2222    2222    2222|
|           |2222  2222  2222  2222  2222  2222  2222  2222                                        2222                        2222    2222    2222|
|           |2222  1111  1111  2222  2222  2222  2222  2222                                        2222                        2222    2222    2222|
|           |2222  1111  1111  1111  1111  1111  2222  2222                                        2222                        2222    1111    2222|
|0:50:00.00 +2222  1111  1111  1111  1111  1111  1111  2222                                        2222  3333              3333  2222    1111    2222|
|           |1111  1111  1111  1111  1111  1111  1111  2222                                        2222  3333              3333  1111    1111    2222|
|           |1111  1111  1111  1111  1111  1111  1111  1111                                        2222  3333              3333  1111    1111    1111|
|           |1111  1111  1111  1111  1111  1111  1111  1111                                        2222  3333              3333  1111    1111    1111|
|0:33:20.00 +1111  1111  1111  1111  1111  1111  1111  1111                                        1111  2222              2222  1111    1111    1111|
|           |1111  1111  1111  1111  1111  1111  1111  1111                              3333      1111  2222              2222  1111    1111    1111|
|           |1111  1111  1111  1111  1111  1111  1111  1111  3333  3333  3333  3333      1111  2222              2222  1111    1111    1111|
|           |1111  1111  1111  1111  1111  1111  1111  1111  3333  3333  3333  2222      1111  2222  3333        2222  1111    1111    1111|
|0:16:40.00 +1111  1111  1111  1111  1111  1111  1111  1111  2222  2222  2222  2222      1111  1111  3333        1111  1111    1111    1111|
|           |1111  1111  1111  1111  1111  1111  1111  1111  2222  2222  2222  2222      1111  1111  2222  3333  1111  1111    1111    1111|
|           |1111  1111  1111  1111  1111  1111  1111  1111  1111  1111  1111  1111      1111  1111  2222  2222  1111  1111    1111    1111|
|           |1111  1111  1111  1111  1111  1111  1111  1111  1111  1111  1111  1111      1111  1111  1111  1111  1111  1111    1111    1111|
|           -------------------------------------------------------------------------------------------------------------------------------|
|              0     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19 |
|                                                          HOUR     Hour Of Day                                                          |
|                                                                                                                                         |
|                                          SYMBOL MODCPU     SYMBOL MODCPU     SYMBOL MODCPU                                            |
|                                            1         1       2         2       3         3                                            |
|                                                                                                                                         |
|                                                                                                                                         |
|                                            Figure 3-208.   Available CPU Cycles                                                         |
+-----------------------------------------------------------------------------------------------------------------------------------------+
```

Overall Page Fault Rate

INQUIRY: DEXPD8 RUN DATE: DDMMMYY

SYSID=VMS1 Month=7 Day=17
BAR CHART OF SPRPSPFL

SPRPSPFL

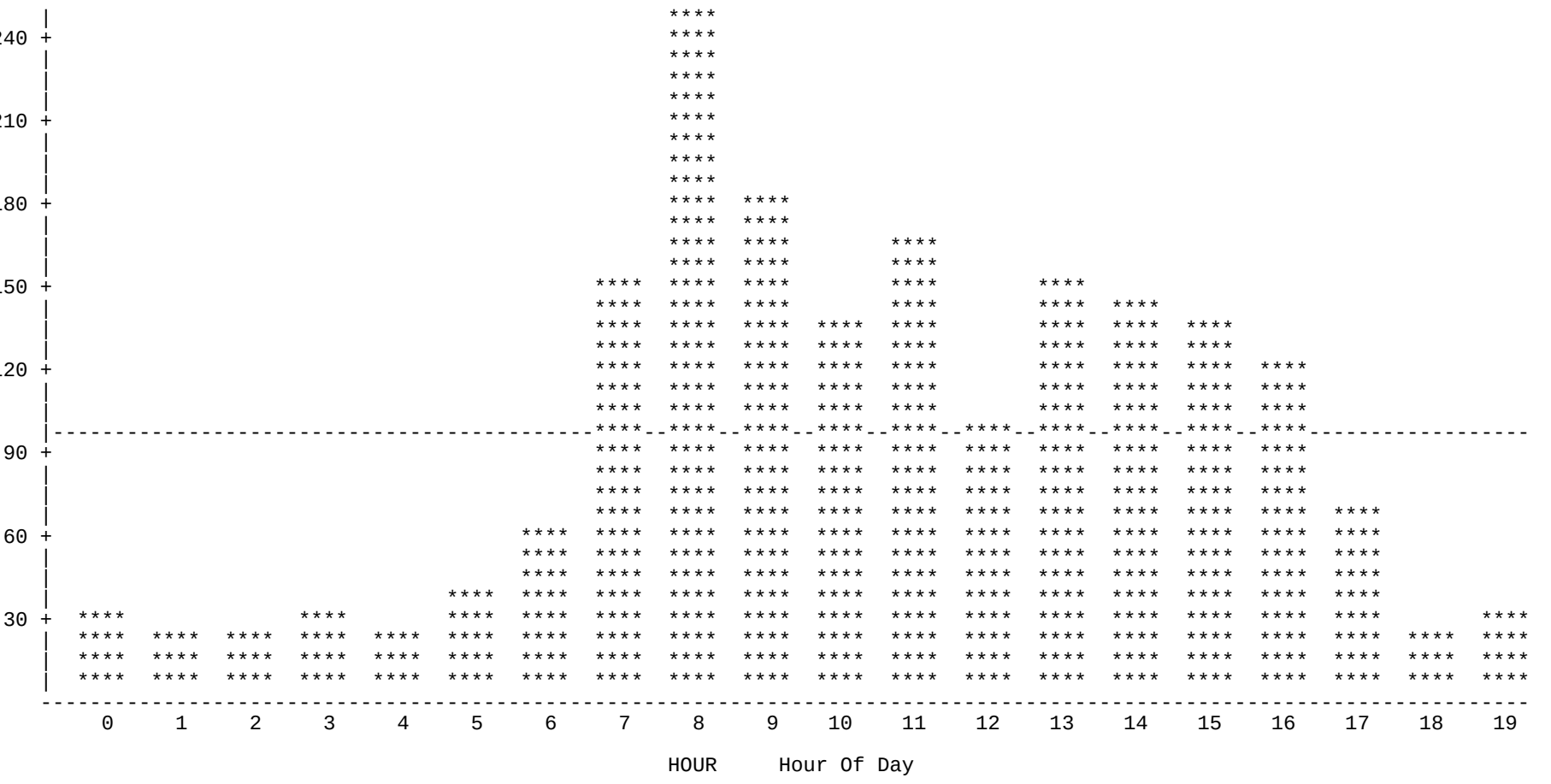

Figure 3-209. Overall Page Fault Rate

Daily System Status - Uptime & Memory Use

INQUIRY: DEXLDH RUN DATE: 24JUL08

System Identifier = VMS2 Hardware = VAX 8350 Year = 08 Month = 7

Day of Month	System Uptime	Initial NonPgDyn (NPAGEDYN)	Maximum NonPgDyn (NPAGEVIR)	Initial SRP Pkts (SRPCOUNT)	Maximum SRP Pkts (SRPCOUNTV)	Initial IRP Pkts (IRPCOUNT)	Maximum IRP Pkts (IRPCOUNTV)	Initial LRP Pkts (LRPCOUNT)	Maximum LRP Pkts (LRPCOUNTV)
9	8:02:42.00	572,416	1,177,600	1,115	3,345	794	2,382	75	225
10	32:02:42.00	572,416	1,177,600	1,115	3,345	794	2,382	75	225
11	56:02:42.00	572,416	1,177,600	1,115	3,345	794	2,382	75	225

Figure 3-210. Daily System Status - Uptime & Memory Use

Daily System Summary - CPU Hourly Averages

INQUIRY: DEXLDI RUN DATE: 24JUL08

SYSID VMS1

Date	Hour	Avg CPU Percent Busy	Kernel+ Int Stack Pct CPU	User Mode Pct CPU	FreeLst Pag Flt Rate	Mod Lst Pag Flt Rate	System Pag Flt Rate	Inswap Rate	Page Fault Rate	Page Read IO Rate	Direct IO Rate	Buffered IO Rate
17JUL2008	0	37.05 %	21.61 %	72.83 %	4.20	2.49	0.05	0.00	30.85	0.70	27.04	6.35
17JUL2008	1	5.81 %	69.96 %	10.77 %	3.64	1.58	0.00	0.00	20.89	0.68	9.30	5.07
17JUL2008	2	8.40 %	57.24 %	24.92 %	3.85	1.93	0.02	0.00	23.66	0.78	10.68	6.08
17JUL2008	3	10.55 %	61.20 %	17.62 %	6.79	2.03	0.09	0.00	26.59	0.86	15.56	7.67
17JUL2008	4	7.96 %	65.36 %	20.52 %	3.96	1.57	0.04	0.00	21.15	0.63	9.45	9.29
17JUL2008	5	12.22 %	64.19 %	17.47 %	6.84	3.92	0.20	0.00	38.21	1.71	13.64	17.81
17JUL2008	6	20.04 %	54.30 %	26.71 %	8.05	4.18	0.25	0.00	60.95	1.50	12.62	16.82
17JUL2008	7	39.40 %	57.27 %	25.27 %	13.83	8.36	0.75	0.00	150.49	3.08	21.98	48.84
17JUL2008	8	86.98 %	52.07 %	30.16 %	16.72	14.10	1.31	0.00	248.65	4.60	41.26	121.89
17JUL2008	9	85.37 %	52.83 %	30.44 %	15.59	14.63	1.67	0.00	177.71	3.20	40.72	115.03
17JUL2008	10	85.65 %	51.40 %	32.46 %	12.49	12.55	2.18	0.00	135.84	3.21	43.89	164.43
17JUL2008	11	83.26 %	52.60 %	31.37 %	17.85	15.37	1.95	0.00	166.12	4.14	40.10	141.04
17JUL2008	12	48.31 %	55.44 %	30.30 %	9.62	9.22	0.94	0.00	101.18	2.46	20.57	95.99
17JUL2008	13	71.99 %	53.23 %	30.99 %	14.74	12.73	1.60	0.00	152.37	3.18	32.01	139.89
17JUL2008	14	89.75 %	50.56 %	34.13 %	15.62	14.05	1.91	0.00	143.03	3.05	39.79	176.31
17JUL2008	15	90.77 %	52.11 %	31.77 %	12.74	12.96	1.87	0.00	133.84	2.86	38.08	193.79
17JUL2008	16	71.28 %	52.77 %	30.78 %	12.97	11.60	1.30	0.00	120.55	3.04	32.61	119.89
17JUL2008	17	27.88 %	50.81 %	35.75 %	10.98	7.30	0.61	0.00	70.14	1.66	11.42	47.83
17JUL2008	18	18.03 %	46.34 %	33.51 %	3.89	2.22	0.06	0.00	24.99	0.75	8.81	19.98
17JUL2008	19	35.07 %	44.79 %	45.45 %	3.83	4.67	0.17	0.00	29.69	0.84	38.91	32.08
17JUL2008	20	61.61 %	54.19 %	33.99 %	5.16	3.03	0.12	0.00	45.05	1.02	81.00	61.24
17JUL2008	21	52.46 %	57.23 %	33.12 %	5.47	7.87	0.17	0.00	44.90	1.19	76.01	60.61
17JUL2008	22	26.34 %	63.30 %	17.47 %	4.21	2.20	0.05	0.00	26.65	0.88	38.38	21.63
17JUL2008	23	29.73 %	52.72 %	32.41 %	4.35	11.85	0.07	0.00	44.70	0.86	27.49	7.69

Figure 3-211. Daily System Summary - CPU Hourly Averages

Daily System Summary - Average Disk Device Statistics

INQUIRY: DEXLDI RUN DATE: 24JUL08

SYSID VMS2

Date	Disk Device	Device Type	Blocks Total	Blocks Free	Pct of Space Used	Disk Avg Response Time	Disk Avg IO Rate	Volume Name	Volume Set Name	Last Dev Err Count
08JUL2008	DUA0	RA81	891,072	43,134	95.15 %	.	.	VMSRL5		0
08JUL2008	DUA1	RA81	891,072	642,231	27.92 %	.	.	VMS2SUSER1		0
08JUL2008	DUA2	RA70	1,133,160	1,003,698	11.42 %	.	.	DISK2		0
										0
09JUL2008	DUA0	RA81	891,072	48,480	94.55 %	0.023	0.253	VMSRL5		4
09JUL2008	DUA1	RA81	891,072	640,812	28.08 %	0.026	0.027	VMS2SUSER1		0
09JUL2008	DUA2	RA70	1,133,160	1,003,698	11.42 %	.	0.000	DISK2		0
										4
10JUL2008	DUA0	RA81	891,072	43,842	95.07 %	0.028	0.101	VMSRL5		4
10JUL2008	DUA1	RA81	891,072	639,447	28.23 %	0.026	0.082	VMS2SUSER1		0
10JUL2008	DUA2	RA70	1,133,160	1,003,698	11.42 %	0.038	0.005	DISK2		0
										4
11JUL2008	DUA0		.	.	.	0.032	1.857			.
11JUL2008	DUA1		.	.	.	0.036	1.759			.
11JUL2008	DUA2		.	.	.	0.042	0.500			.
										0

Figure 3-212. Daily System Summary - Average Disk Device Statistics

Weekly Average Disk Response Time

INQUIRY: DEXPW2 RUN DATE: 24JUL08

SYSID=VMS2 Disk Device=DUA0 Year=08
BAR CHART OF DSKAVRES

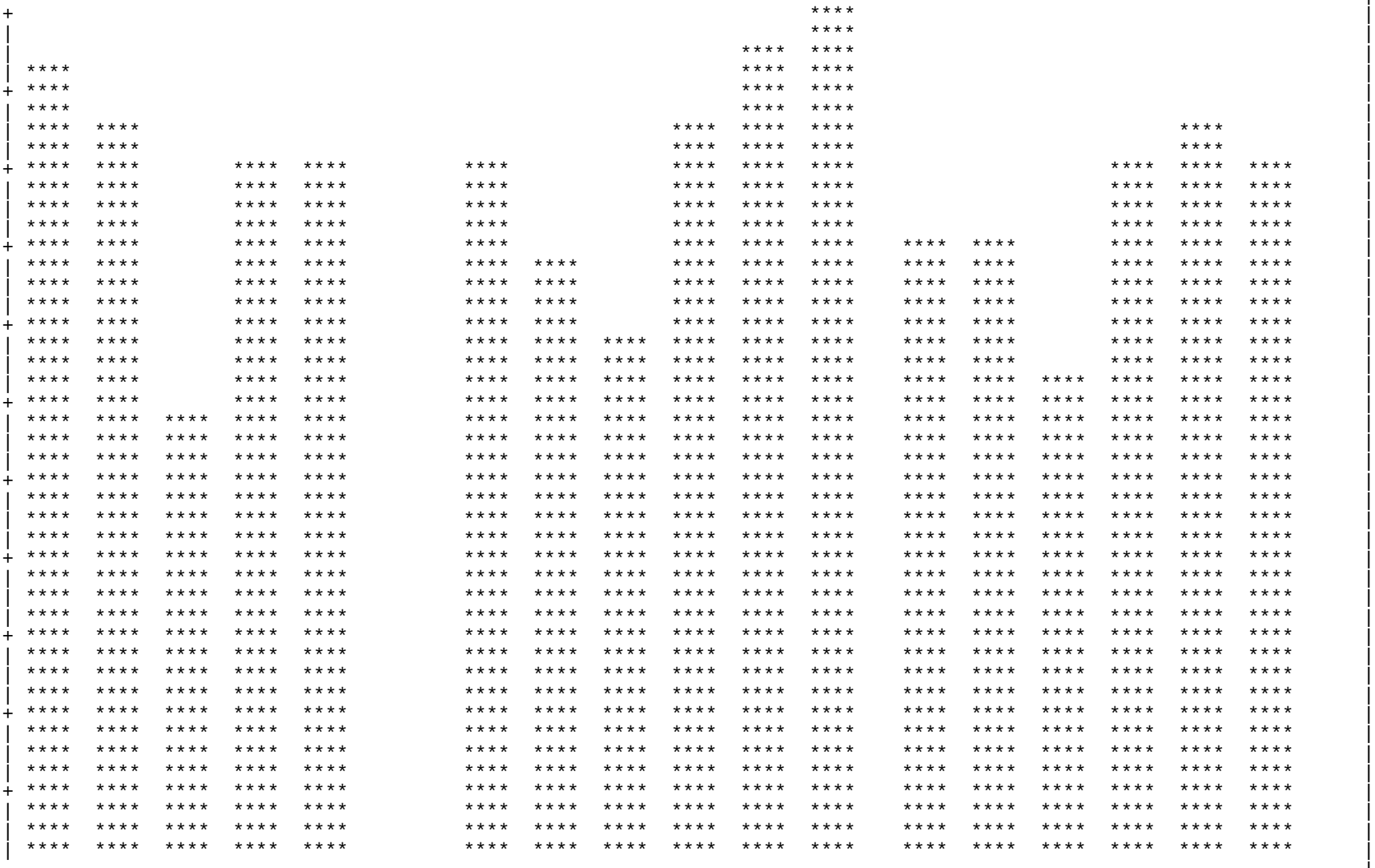

Figure 3-213. Weekly Average Disk Response Time

Service Performed by the System - CPU Avg

INQUIRY: DEXPW4 RUN DATE: DDJULYY

SYSID=VMS1

PLOT OF SRVPCSYS*HOUR SYMBOL USED IS *

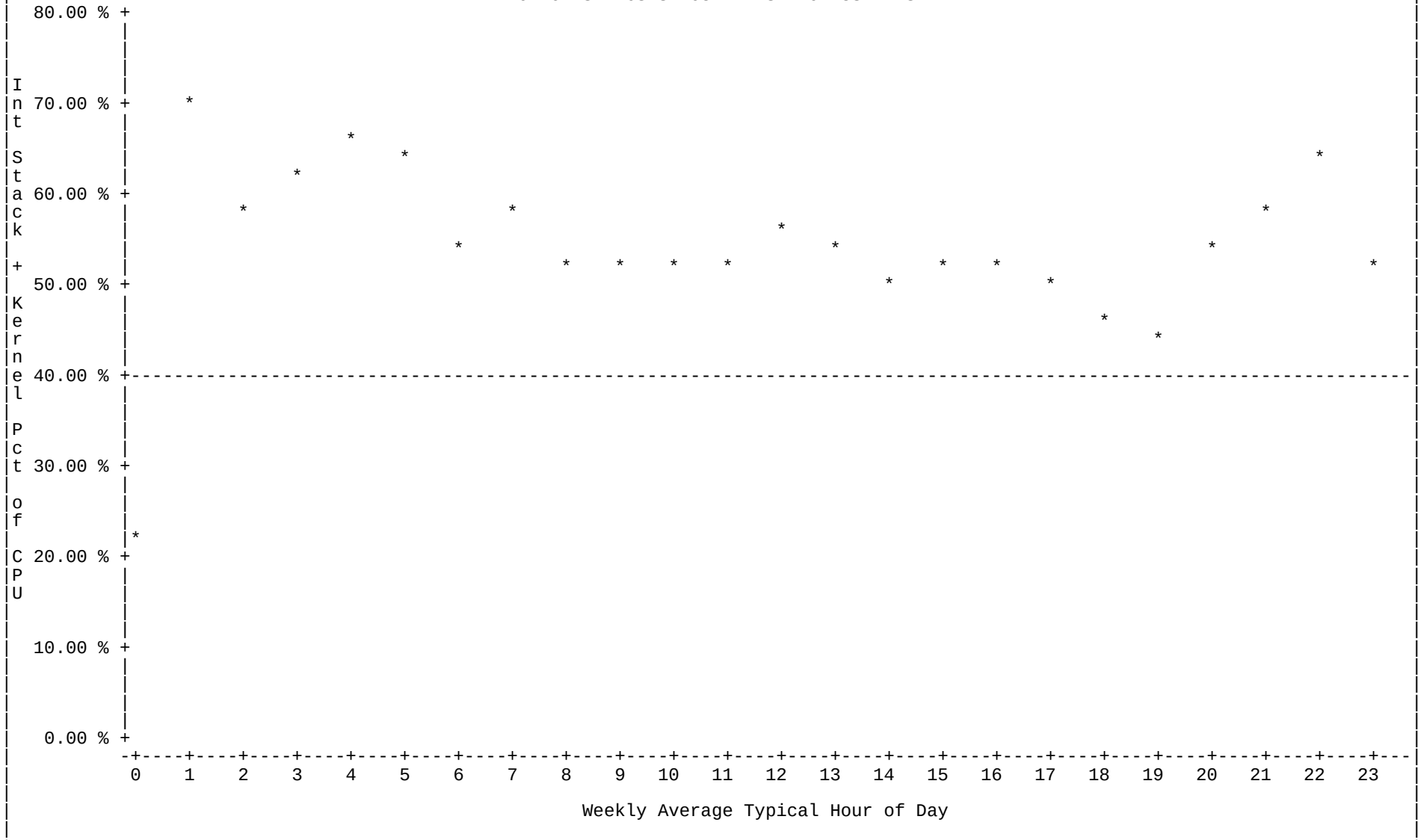

Figure 3-214. Service Performed by the System - CPU Avg

3.1.3 - VAX/VMS Disk Storage Analysis Inquiries

This section lists the MICF inquiries for VAX/VMS disk storage analysis. A brief description is given for each CA MICS VAX/VMS Analyzer inquiry in this category. Sample report output follows this section.

DEXLXA - Detail Directory Top Level Space

Sample: Fig 3-301.

This inquiry is useful in determining overall disk space consumption patterns across all disk devices on each VMS system. The reports list file counts, block usage, occupancy, and other statistics for each directory top level. The directory top level is defined as the first node of directory name, up to 12 characters in length. This grouping of directories gives a compact summary of disk usage while retaining enough information to identify the user or owner of the files represented.

DEXLXB - Detail Directory Top Level Ranking (2 Reports) Space Allocated & Unused

Sample: Fig 3-302 and 303.

This inquiry gives you a quick way to determine who is using and potentially wasting the most disk space, across all disk devices on each VMS system. The report shows the top 40 disk space users based on blocks allocated and on unused blocks. The reports list file counts, block usage, occupancy, and other statistics for each directory top level. The directory top level is defined as the first node of directory name, up to 12 characters in length. This grouping of directories gives a compact summary of disk usage while retaining enough information to identify the main users or owners of the files represented.

DEXLXC - Detail Directory Structure and Space

Sample: Fig 3-304.

This inquiry shows you the directory structures and disk space consumption patterns for each disk device on each VMS system. The reports list file counts, block usage, occupancy, and other statistics for each directory on each disk device. This report is a useful reference for understanding the file allocation and space usage pattern on an individual disk, to a greater degree of detail than directory top level.

DEXLXD - Detail File Space Ranking - Allocated

Sample: Fig 3-305.

This inquiry shows you where the largest files are on each disk device on each VMS system. The report lists the top 40 largest files based on blocks allocated and includes blocks used and unused, percent of space used, and occupancy in megabyte hours for each individual file in the ranking list. This report is useful for finding and potentially reclaiming large areas of space, and for understanding where the majority of disk space usage is concentrated on each disk device.

DEXLXE - Detail Disk Quota Statistics

Sample: Fig 3-306.

This inquiry lists the disk quota statistics for each quota-enabled disk measured by the CA MICS system usage collection routine. In addition to the standard output produced by DISKQUOTA SHOW, the report lists disk usage occupancy in megabyte hours, and percent of quota used, based on the disk quota readings. This report can help you manage quota settings and anticipate the need for quota increases, before overdraft limits are exceeded.

DEXLDA - Daily Directory Top Level Usage

Sample: Fig 3-307.

This inquiry is useful in determining overall disk space consumption patterns across all disk devices on each VMS system. The reports list disk space occupancy statistics for each directory top level. The directory top level is defined as the first node of directory name, up to 12 characters in length. This grouping of directories gives a compact summary of disk usage while retaining enough information to identify the user or owner of the files represented. This report is based on the DESDKU file in

the DAYS timespan, which is a much smaller file than the DETAIL level. Thus, it is useful for reviewing disk space consumption in reasonable detail over a longer period of time than the DETAIL timespan could reasonably support.

DEXLDB - Daily Directory Top Level Ranking:
(2 Reports) Megabyte Hours Allocated & Unused

Sample: Fig 3-308 and 309.

This inquiry gives you a quick way to determine who is using and potentially wasting the most disk space, across all disk devices on each VMS system. The report shows the top 40 disk space users based on allocated and unused occupancy, listing occupancy statistics for each directory top level. The directory top level is defined as the first node of directory name, up to 12 characters in length. This report is based on the DESDKU file in the DAYS timespan, which is a much smaller file than the DETAIL level. Thus, it is useful for reviewing disk space consumption in reasonable detail over a longer period of time than the DETAIL timespan could reasonably support.

DEXPWF - Weekly Disk Device Free Space
DEXCWF

Sample: Fig 3-310.

This inquiry shows you the relative amount of free space on each disk device on each VMS system, in the form of a horizontal bar graph. Each bar represents the free space measurement for a single disk in a given week. This report can be used to compare many disks for a single week or for spotting overall disk space consumption trends over many weeks, based on the cycle selection you choose at report execution time.

Note: In color graphics mode, you may want to customize this inquiry to fit either many weeks or many devices on one chart. In printer graphics mode, this report will continue over multiple pages.

DEXPWG - Weekly Disk Device Space Utilization
DEXCWG

Sample: Fig 3-311.

This inquiry shows you the disk space percent utilization of each disk device on each VMS system, in the form of a horizontal bar graph. Each bar represents the percent of total space in use for a single disk in a given week. This report can be used to compare many disks for a single week or can used for spotting overall disk space consumption trends over many weeks, based on the cycle selection you choose at report execution time.

Note: In color graphics mode, you may want to customize this inquiry to fit either many weeks or many devices on one chart. In printer graphics mode, this report will continue over multiple pages.

DEXLMA - Monthly Disk Usage Summary

Sample: Fig 3-312.

This inquiry is useful in determining overall disk space consumption patterns across all disk devices on each VMS system, based on organizational account code levels. The reports list disk space occupancy statistics for the first three levels of account codes defined to the VAX/VMS Analyzer. This report helps you relate disk space usage requirements to business activities and organizational units. Based on the DESDKU file in the MONTHS timespan, the report can cover disk usage over long periods of time.

DEXLMB - Monthly Disk Usage Ranking:
(2 Reports) Megabyte Hours Allocated & Unused

Sample: Fig 3-313 and 314.

This inquiry gives you a way to see who is using and potentially wasting the most disk space within each organizational unit at your installation. The reports list disk space occupancy statistics for the first three levels of account codes, ranked at the third level of account code, based on allocated and unused occupancy. This report can help groups within the organization to manage their use of space more effectively by reviewing relative space requirements for reasonable levels of consumption.

Detail Directory Top Level Space

INQUIRY: DEXLXA RUN DATE: DDMMMYY

System Identifier = VMS2 Date = DDMMMYY:00:00

Directory Top Level	Disk Device Name	File Count	Blocks (512) Allocated	Blocks (512) Used	Blocks (512) Unused	Percent Space Used	Avg File Blocks Allocated	Megabyte Hours Allocated
ACCTING	DUA1	3	15	9	6	60.00 %	5	0.059
AD9SF70	DUA0	1	3	3	0	100.00 %	3	0.012
AD9WD06	DUA0	1	3	3	0	100.00 %	3	0.012
AHZEBEL	DUA1	16	99	83	16	83.83 %	6	0.387
ALLIN1	DUA0	1,249	58,209	56,343	1,866	96.79 %	47	227.379
BACKUP	DUA0	8	21	10	11	47.61 %	3	0.082
BACKUP	DUA2	5	18	5	13	27.77 %	4	0.070
BBBTTAGL	DUA1	2	6	2	4	33.33 %	3	0.023
BPPRTER	DUA1	2	6	2	4	33.33 %	3	0.023
CD9JE15	DUA0	30	366	300	66	81.96 %	12	1.430
CLASS1	DUA0	14	2,271	2,256	15	99.33 %	162	8.871
CLASS2	DUA0	4	12	6	6	50.00 %	3	0.047
CLASS3	DUA0	1	3	3	0	100.00 %	3	0.012
CLASS4	DUA0	1	3	3	0	100.00 %	3	0.012
CLASS5	DUA0	1	3	3	0	100.00 %	3	0.012
CSTTRNER	DUA1	2	6	2	4	33.33 %	3	0.023
CS9RG99	DUA0	24	135	104	31	77.03 %	6	0.527
DBDBD	DUA1	4	12	12	0	100.00 %	3	0.047
DD9EC17	DUA0	26	303	238	65	78.54 %	12	1.184
DD9GW20	DUA0	66	918	775	143	84.42 %	14	3.586
DD9JB92	DUA0	47	6,921	6,632	289	95.82 %	147	27.035
DD9JK54	DUA0	16	5,139	5,130	9	99.82 %	321	20.074
DD9KL46	DUA0	5	15	15	0	100.00 %	3	0.059
DD9RV54	DUA0	44	8,286	8,189	97	98.82 %	188	32.367
DD9WH74	DUA0	3	9	9	0	100.00 %	3	0.035
DIABSPKER	DUA0	3	315	308	7	97.77 %	105	1.230
DSDCTUTE	DUA1	2	6	2	4	33.33 %	3	0.023
DSJAG11	DUA0	2	6	5	1	83.33 %	3	0.023
DSJDE11	DUA0	23	294	229	65	77.89 %	13	1.148
DSJGD11	DUA0	25	138	65	73	47.10 %	6	0.539
DSJJZ11	DUA0	25	300	233	67	77.66 %	12	1.172
DSJPK11	DUA0	23	126	58	68	46.03 %	5	0.492
DSJPR11	DUA0	23	132	64	68	48.48 %	6	0.516
DSJTM11	DUA0	23	126	58	68	46.03 %	5	0.492
FALAPTCEI	DUA2	34	65,481	65,431	50	99.92 %	1,926	255.785
FLEX	DUA1	167	128,562	128,307	255	99.80 %	770	502.195
FVVT	DUA0	1	9	9	0	100.00 %	9	0.035
HCCZBILL	DUA1	2	6	2	4	33.33 %	3	0.023
JCOMMINS	DUA1	2	6	2	4	33.33 %	3	0.023
JCULQMAN	DUA1	3	9	3	6	33.33 %	3	0.035
JELJLJ	DUA1	2	6	2	4	33.33 %	3	0.023
JFQ	DUA2	5	66	61	5	92.42 %	13	0.258

Figure 3-301. Detail Directory Top Level Space

Detail Directory Top Level Ranking - Allocated

INQUIRY: DEXLXB RUN DATE: DDMMMYY

System Identifier = VMS2 Date = DDMMMYY:00:00

Blocks (512) Allocated	Directory Top Level	Disk Device Name	File Count	Blocks (512) Used	Blocks (512) Unused	Percent Space Used	Avg File Blocks Allocated	Megabytes Hours Allocated
729,002	SYS0	DUA0	2,962	701,128	27,874	96.17 %	246	2847.664
128,562	FLEX	DUA1	167	128,307	255	99.80 %	770	502.195
65,481	FALAPTCEI	DUA2	34	65,431	50	99.92 %	1,926	255.785
63,870	JQINJQ	DUA1	566	62,997	873	98.63 %	113	249.492
58,209	ALLIN1	DUA0	1,249	56,343	1,866	96.79 %	47	227.379
40,971	OWDCLE	DUA2	4	40,963	8	99.98 %	10,243	160.043
36,339	PATLM	DUA1	409	35,742	597	98.35 %	89	141.949
22,740	000000	DUA2	20	22,298	442	98.05 %	1,137	88.828
16,464	METANET	DUA0	277	15,939	525	96.81 %	59	64.313
11,541	000000	DUA1	65	11,410	131	98.86 %	178	45.082
8,286	DD9RV54	DUA0	44	8,189	97	98.82 %	188	32.367
6,921	DD9JB92	DUA0	47	6,632	289	95.82 %	147	27.035
6,639	000000	DUA0	74	6,505	134	97.98 %	90	25.934
6,618	VMS4	DUA0	47	6,486	132	98.00 %	141	25.852
5,703	SANMN	DUA1	11	5,692	11	99.80 %	518	22.277
5,139	DD9JK54	DUA0	16	5,130	9	99.82 %	321	20.074
3,300	OPERATOR	DUA0	37	3,243	57	98.27 %	89	12.891
2,271	CLASS1	DUA0	14	2,256	15	99.33 %	162	8.871
1,686	OWDFIN	DUA1	68	1,585	101	94.00 %	25	6.586
918	DD9GW20	DUA0	66	775	143	84.42 %	14	3.586
759	SYSE	DUA0	7	752	7	99.07 %	108	2.965
642	KERMIT	DUA1	8	614	28	95.63 %	80	2.508
546	PRODUCTS	DUA1	12	530	16	97.06 %	46	2.133
366	CD9JE15	DUA0	30	300	66	81.96 %	12	1.430
315	DIABSPKER	DUA0	3	308	7	97.77 %	105	1.230
312	MD1MP41	DUA0	26	245	67	78.52 %	12	1.219
303	DD9EC17	DUA0	26	238	65	78.54 %	12	1.184
300	DSJJZ11	DUA0	25	233	67	77.66 %	12	1.172
294	DSJDE11	DUA0	23	229	65	77.89 %	13	1.148
273	PRODUCTS	DUA0	3	265	8	97.06 %	91	1.066
174	REGHIJBIONS	DUA0	2	169	5	97.12 %	87	0.680
144	MA9KM38	DUA0	27	70	74	48.61 %	5	0.563
138	DSJGD11	DUA0	25	65	73	47.10 %	6	0.539
135	CS9RG99	DUA0	24	104	31	77.03 %	6	0.527
132	DSJPR11	DUA0	23	64	68	48.48 %	6	0.516
126	DSJPK11	DUA0	23	58	68	46.03 %	5	0.492
126	DSJTM11	DUA0	23	58	68	46.03 %	5	0.492
126	STXDEMO0	DUA0	23	59	67	46.82 %	5	0.492
105	JVOLM	DUA1	10	90	15	85.71 %	11	0.410
99	AHZEBEL	DUA1	16	83	16	83.83 %	6	0.387

Figure 3-302. Detail Directory Top Level Ranking - Allocated

Detail Directory Top Level Ranking - Unused

INQUIRY: DEXLXB RUN DATE: 24JUL08

System Identifier = VMS2 Date = 09JUL08:00:00

Blocks (512) Unused	Directory Top Level	Disk Device Name	File Count	Blocks (512) Allocated	Blocks (512) Used	Percent Space Used	Avg File Blocks Allocated	Megabytes Hours Allocated
27,874	SYS0	DUA0	2,962	729,002	701,128	96.17 %	246	2847.664
1,866	ALLIN1	DUA0	1,249	58,209	56,343	96.79 %	47	227.379
873	JQINJQ	DUA1	566	63,870	62,997	98.63 %	113	249.492
597	PATLM	DUA1	409	36,339	35,742	98.35 %	89	141.949
525	METANET	DUA0	277	16,464	15,939	96.81 %	59	64.313
442	000000	DUA2	20	22,740	22,298	98.05 %	1,137	88.828
289	DD9JB92	DUA0	47	6,921	6,632	95.82 %	147	27.035
255	FLEX	DUA1	167	128,562	128,307	99.80 %	770	502.195
143	DD9GW20	DUA0	66	918	775	84.42 %	14	3.586
134	000000	DUA0	74	6,639	6,505	97.98 %	90	25.934
132	VMS4	DUA0	47	6,618	6,486	98.00 %	141	25.852
131	000000	DUA1	65	11,541	11,410	98.86 %	178	45.082
101	OWDFIN	DUA1	68	1,686	1,585	94.00 %	25	6.586
97	DD9RV54	DUA0	44	8,286	8,189	98.82 %	188	32.367
74	MA9KM38	DUA0	27	144	70	48.61 %	5	0.563
73	DSJGD11	DUA0	25	138	65	47.10 %	6	0.539
68	DSJPK11	DUA0	23	126	58	46.03 %	5	0.492
68	DSJPR11	DUA0	23	132	64	48.48 %	6	0.516
68	DSJTM11	DUA0	23	126	58	46.03 %	5	0.492
67	DSJJZ11	DUA0	25	300	233	77.66 %	12	1.172
67	MD1MP41	DUA0	26	312	245	78.52 %	12	1.219
67	STXDEMO0	DUA0	23	126	59	46.82 %	5	0.492
66	CD9JE15	DUA0	30	366	300	81.96 %	12	1.430
65	DD9EC17	DUA0	26	303	238	78.54 %	12	1.184
65	DSJDE11	DUA0	23	294	229	77.89 %	13	1.148
64	PERFORM	DUA2	17	87	23	26.43 %	5	0.340
57	OPERATOR	DUA0	37	3,300	3,243	98.27 %	89	12.891
50	FALAPTCEI	DUA2	34	65,481	65,431	99.92 %	1,926	255.785
31	CS9RG99	DUA0	24	135	104	77.03 %	6	0.527
28	KERMIT	DUA1	8	642	614	95.63 %	80	2.508
16	AHZEBEL	DUA1	16	99	83	83.83 %	6	0.387
16	PRODUCTS	DUA1	12	546	530	97.06 %	46	2.133
15	CLASS1	DUA0	14	2,271	2,256	99.33 %	162	8.871
15	JVOLM	DUA1	10	105	90	85.71 %	11	0.410
13	BACKUP	DUA2	5	18	5	27.77 %	4	0.070
11	BACKUP	DUA0	8	21	10	47.61 %	3	0.082
11	SANMN	DUA1	11	5,703	5,692	99.80 %	518	22.277
9	DD9JK54	DUA0	16	5,139	5,130	99.82 %	321	20.074
9	PRINTERSS	DUA1	3	18	9	50.00 %	6	0.070
8	PRODUCTS	DUA0	3	273	265	97.06 %	91	1.066

Figure 3-303. Detail Directory Top Level Ranking - Unused

Detail Directory Structure and Space

INQUIRY: DEXLXC RUN DATE: DDMMMYY

System Identifier = VMS2 Date = DDMMMYY:00:00 Device Name = DUA0

File Count	Blocks (512) Allocated	Blocks (512) Used	Blocks (512) Unused	Percent Space Used	Avg File Blocks Allocated	Megabyte Hours Allocated	Directory Name
1	3	3	0	100.00 %	3	0.012	AD9SF70
1	3	3	0	100.00 %	3	0.012	AD9WD06
65	345	257	88	74.49 %	5	1.348	ALLIN1
22	78	46	32	58.97 %	4	0.305	ALLIN1.BLP_ENGLISH
7	21	8	13	38.09 %	3	0.082	ALLIN1.BLP_SHARE
167	2,736	2,573	163	94.04 %	16	10.688	ALLIN1.CBI_ENGLISH
9	132	132	0	100.00 %	15	0.516	ALLIN1.DATA_ENGLISH
78	3,090	2,985	105	96.60 %	40	12.070	ALLIN1.DATA_SHARE
24	177	145	32	81.92 %	7	0.691	ALLIN1.DO_ENGLISH
83	528	445	83	84.28 %	6	2.063	ALLIN1.DO_SHARE
49	780	685	95	87.82 %	16	3.047	ALLIN1.IVP_ENGLISH
26	312	281	31	90.06 %	12	1.219	ALLIN1.IVPUSER
160	15,006	14,786	220	98.53 %	94	58.617	ALLIN1.LIB_ENGLISH
330	12,960	12,643	317	97.55 %	39	50.625	ALLIN1.LIB_SHARE
10	87	48	39	55.17 %	9	0.340	ALLIN1.LIB_SHARE.USER
18	4,128	4,107	21	99.49 %	229	16.125	ALLIN1.LLV_ENGLISH
18	1,128	1,112	16	98.58 %	63	4.406	ALLIN1.LOG
27	324	294	30	90.74 %	12	1.266	ALLIN1.MGR
4	9	7	2	77.77 %	2	0.035	ALLIN1.MGR.DOC0
1	3	3	0	100.00 %	3	0.012	ALLIN1.MGR.DOC1
2	15	13	2	86.66 %	8	0.059	ALLIN1.MGR.DOC3
1	6	4	2	66.66 %	6	0.023	ALLIN1.MGR.DOC5
1	3	3	0	100.00 %	3	0.012	ALLIN1.MGR.DOC7
1	3	3	0	100.00 %	3	0.012	ALLIN1.MGR.DOC8
1	3	3	0	100.00 %	3	0.012	ALLIN1.MGR.DOC9
2	9	7	2	77.77 %	5	0.035	ALLIN1.MGR.MSG
22	123	96	27	78.04 %	6	0.480	ALLIN1.POSTMASTE
34	507	458	49	90.33 %	15	1.980	ALLIN1.PRINT
2	6	4	2	66.66 %	3	0.023	ALLIN1.SCP_ENGLISH
1	3	1	2	33.33 %	3	0.012	ALLIN1.SCP_SHARE
1	114	114	0	100.00 %	114	0.445	ALLIN1.SHARED_E
8	42	35	7	83.33 %	5	0.164	ALLIN1.SHARE1
14	42	14	28	33.33 %	3	0.164	ALLIN1.SITE
1	3	2	1	66.66 %	3	0.012	ALLIN1.SITE.DEV_ENGLISH
6	18	11	7	61.11 %	3	0.070	ALLIN1.SITE.LIB_ENGLISH
54	15,468	15,018	450	97.09 %	286	60.422	ALLIN1.SOURCES
4	12	5	7	41.66 %	3	0.047	BACKUP
7	51	49	2	96.07 %	7	0.199	CD9JE15
22	291	228	63	78.35 %	13	1.137	CD9JE15.A1
1	24	23	1	95.83 %	24	0.094	CD9JE15.A1.DOC1
14	2,271	2,256	15	99.33 %	162	8.871	CLASS1

Figure 3-304. Detail Directory Structure and Space

Detail File Space Ranking - Allocated

INQUIRY: DEXLXD RUN DATE: DDMMMYY

System Identifier = VMS2 Date = DDMMMYY:00:00 Device Name = DUA0

Blocks (512) Allocated	Blocks (512) Used	Blocks (512) Unused	Percent Space Used	Megabytes Hours Allocated	File Specification
90,000	90,000	0	100.00 %	351.563	<SYS0.SYSEXE>PAGEFILE.SYS;1
65,541	65,541	0	100.00 %	256.020	<SYS0.SYSEXE>SYSDUMP.DMP;1
44,953	44,950	3	99.99 %	175.598	<SYS0.SYSCOMMON.MONARCHIVE>MON_SEND.DAT;2
44,952	38,173	6,779	84.91 %	175.594	<SYS0.SYSCOMMON.MONARCHIVE>MON_SEND.DAT;4
44,952	44,950	2	99.99 %	175.594	<SYS0.SYSCOMMON.MONARCHIVE>MONITOR.DAT;2
27,765	27,764	1	99.99 %	108.457	<SYS0.SYSCOMMON.MONARCHIVE>MONITOR.DAT;4
27,765	10,944	16,821	39.41 %	108.457	<SYS0.SYSCOMMON.MONARCHIVE>MONITOR.DAT;5
19,953	19,953	0	100.00 %	77.941	<SYS0.SYSCOMMON.MONARCHIVE>MON_SEND.DAT;3
19,953	19,953	0	100.00 %	77.941	<SYS0.SYSCOMMON.MONARCHIVE>MONITOR.DAT;3
15,804	15,801	3	99.98 %	61.734	<SYS0.SYSCOMMON.MONARCHIVE>VAX1_MON30MAY.DAT;1
14,114	14,112	2	99.98 %	55.133	<SYS0.SYSCOMMON.MONARCHIVE>VAX1_MON1JUL.DAT;1
13,725	13,720	5	99.96 %	53.613	<SYS0.SYSCOMMON.MONARCHIVE>VAX1_MON11MAY.DAT;1
11,898	11,898	0	100.00 %	46.477	<SYS0.SYSEXE>SPM033.B;1
11,531	11,524	7	99.93 %	45.043	<SYS0.SYSCOMMON.MONARCHIVE>VAX1_MON13JUN.DAT;1
10,945	10,944	1	99.99 %	42.754	<SYS0.SYSCOMMON.MONARCHIVE>VAX1_MON17APR.DAT;1
9,402	9,400	2	99.97 %	36.727	<SYS0.SYSEXE>SWAPFILE.SYS;1
6,522	6,520	2	99.96 %	25.477	<SYS0.SYSCOMMON.SYSEXE>SPM$ANALYZER.EXE;1
6,354	6,354	0	100.00 %	24.820	<000000>INDEXF.SYS;1
5,955	5,934	21	99.64 %	23.262	<VMS4.SYSMGR>ACCOUNTNG.DAT;4
5,757	5,757	0	100.00 %	22.488	<ALLIN1.SOURCES>OALIBR.OLB;1
4,995	4,993	2	99.95 %	19.512	<ALLIN1.LIB_SHARE>OA$MAIN.EXE;1
4,695	4,695	0	100.00 %	18.340	<SYS0.SYSCOMMON.CDDPLUS>CDD$DATABASE.RDB;1
4,422	4,420	2	99.95 %	17.273	<SYS0.SYSCOMMON.SYSHLP>HELPLIB.HLB;4
4,296	4,296	0	100.00 %	16.781	<SYS0.SYSCOMMON.SNA$CSV>PGHCT1.DMP;2
3,984	3,982	2	99.94 %	15.563	<SYS0.SYSCOMMON.SYSLIB>STARLET.OLB;4
3,951	3,951	0	100.00 %	15.434	<SYS0.SYSCOMMON.CDD$TEMPLATE>CDD$DATABASE.RDB;1
3,618	3,616	2	99.94 %	14.133	<ALLIN1.LIB_ENGLISH>HLPSOURCE.HLB;1
3,282	3,280	2	99.93 %	12.820	<ALLIN1.LIB_ENGLISH>OAHELP.HLB;1
3,207	3,205	2	99.93 %	12.527	<SYS0.SYSMGR>ALLTEST.DAT;2
3,061	3,059	2	99.93 %	11.957	<SYS0.SYSCOMMON.SYSLIB>SNALU62DF.ADA;2
3,060	3,059	1	99.96 %	11.953	<SYS0.SYSCOMMON.SYSLIB>SNALU62DF.ADA;3
3,030	3,029	1	99.96 %	11.836	<SYS0.SYSCOMMON.SYSLIB>LIB.REQ;1
2,580	2,580	0	100.00 %	10.078	<SYS0.SYSCOMMON.MONITOR>MON_MICS.DAT;7
2,553	2,549	4	99.84 %	9.973	<SYS0.SYSCOMMON.VWYANAGER>VXOMS.EXE;1
2,496	2,496	0	100.00 %	9.750	<ALLIN1.LLV_ENGLISH>OALLV.OLB;1
2,478	2,478	0	100.00 %	9.680	<ALLIN1.DATA_SHARE>CM_SDC_ELEMENTS.DAT;1
2,469	2,468	1	99.95 %	9.645	<SYS0.SYSCOMMON.VWYASTER>VWYASTER.EXE;1
2,379	2,379	0	100.00 %	9.293	<SYS0.SYSCOMMON.SYSLIB>STARLET.REQ;1
2,359	2,356	3	99.87 %	9.215	<SYS0.SYSCOMMON.MONARCHIVE>VAX1_MON20APR.DAT;1
2,329	2,328	1	99.95 %	9.098	<SYS0.SYSMGR>SNATRACE.DAT;6

Figure 3-305. Detail File Space Ranking - Allocated

Detail Disk Quota Statistics

INQUIRY: DEXLXE RUN DATE: DDMMMYY

System Identifier = VMS2 Date = DDMMMYY:00:00 Device Name = DUA2

Disk Quota User ID Code (UIC)	Disk Quota Usage (Blocks)	Permanent Quota (Blocks)	Overdraft Limit (Blocks)	Disk Quota Usage Mbyte Hrs	Usage Percent of Quota
SEABBTO,JQINJQ	76	1,000	100	0.297	7.60 %
SYSTEM	11,238	1,000	100	43.898	1123.80 %
0,0	0	1,000	100	0.000	0.00 %
1,1	54,510	1,000	100	212.930	5451.00 %
10,11	70	1,000	100	0.273	7.00 %
377,4	40,986	1,000	100	160.102	4098.60 %
377,5	8	1,000	100	0.031	0.80 %
377,6	460	1,000	100	1.797	46.00 %
	107,348			419.328	

Figure 3-306. Detail Disk Quota Statistics

Daily Directory Top Level Usage

INQUIRY: DEXLDA RUN DATE: 24JUL08

System Identifier = VMS2 Year = 08 Month = 7 Day = 9

Directory Top Level	Disk Device Name	Megabyte Hours Allocated	Megabyte Hours Used	Megabyte Hours Unused	Megabyte Hours Pct Used
ACCTING	DUA1	0.059	0.035	0.023	60.00 %
AD9SF70	DUA0	0.012	0.012	0.000	100.00 %
AD9WD06	DUA0	0.012	0.012	0.000	100.00 %
AHZEBEL	DUA1	0.387	0.324	0.063	83.83 %
ALLIN1	DUA0	227.379	220.090	7.289	96.79 %
BACKUP	DUA0	0.082	0.039	0.043	47.61 %
BACKUP	DUA2	0.070	0.020	0.051	27.77 %
BBBTTAGL	DUA1	0.023	0.008	0.016	33.33 %
BPPRTER	DUA1	0.023	0.008	0.016	33.33 %
CD9JE15	DUA0	1.430	1.172	0.258	81.96 %
CLASS1	DUA0	8.871	8.813	0.059	99.33 %
CLASS2	DUA0	0.047	0.023	0.023	50.00 %
CLASS3	DUA0	0.012	0.012	0.000	100.00 %
CLASS4	DUA0	0.012	0.012	0.000	100.00 %
CLASS5	DUA0	0.012	0.012	0.000	100.00 %
CSTTRNER	DUA1	0.023	0.008	0.016	33.33 %
CS9RG99	DUA0	0.527	0.406	0.121	77.03 %
DBDBD	DUA1	0.047	0.047	0.000	100.00 %
DD9EC17	DUA0	1.184	0.930	0.254	78.54 %
DD9GW20	DUA0	3.586	3.027	0.559	84.42 %
DD9JB92	DUA0	27.035	25.906	1.129	95.82 %
DD9JK54	DUA0	20.074	20.039	0.035	99.82 %
DD9KL46	DUA0	0.059	0.059	0.000	100.00 %
DD9RV54	DUA0	32.367	31.988	0.379	98.82 %
DD9WH74	DUA0	0.035	0.035	0.000	100.00 %
DIABSPKER	DUA0	1.230	1.203	0.027	97.77 %
DSDCTUTE	DUA1	0.023	0.008	0.016	33.33 %
DSJAG11	DUA0	0.023	0.020	0.004	83.33 %
DSJDE11	DUA0	1.148	0.895	0.254	77.89 %
DSJGD11	DUA0	0.539	0.254	0.285	47.10 %
DSJJZ11	DUA0	1.172	0.910	0.262	77.66 %
DSJPK11	DUA0	0.492	0.227	0.266	46.03 %
DSJPR11	DUA0	0.516	0.250	0.266	48.48 %
DSJTM11	DUA0	0.492	0.227	0.266	46.03 %
FALAPTCEI	DUA2	255.785	255.590	0.195	99.92 %
FLEX	DUA1	502.195	501.199	0.996	99.80 %
FVVT	DUA0	0.035	0.035	0.000	100.00 %
HCCZBILL	DUA1	0.023	0.008	0.016	33.33 %
JCOMMINS	DUA1	0.023	0.008	0.016	33.33 %
JCULQMAN	DUA1	0.035	0.012	0.023	33.33 %
JELJLJ	DUA1	0.023	0.008	0.016	33.33 %
JFQ	DUA2	0.258	0.238	0.020	92.42 %
JGUMB	DUA1	0.012	0.004	0.008	33.33 %
JQINJQ	DUA1	249.492	246.082	3.410	98.63 %

Figure 3-307. Daily Directory Top Level Usage

Daily Directory Top Level Ranking - Allocated

INQUIRY: DEXLDB RUN DATE: 24JUL08

System Identifier = VMS2 Year = 08 Month = 7 Day = 9

Megabyte Hours Allocated	Directory Top Level	Disk Device Name	Megabyte Hours Used	Megabyte Hours Unused	Megabyte Hours Pct Used
2847.664	SYS0	DUA0	2738.781	108.883	96.17 %
502.195	FLEX	DUA1	501.199	0.996	99.80 %
255.785	FALAPTCEI	DUA2	255.590	0.195	99.92 %
249.492	JQINJQ	DUA1	246.082	3.410	98.63 %
227.379	ALLIN1	DUA0	220.090	7.289	96.79 %
160.043	OWDCLE	DUA2	160.012	0.031	99.98 %
141.949	PATLM	DUA1	139.617	2.332	98.35 %
88.828	000000	DUA2	87.102	1.727	98.05 %
64.313	METANET	DUA0	62.262	2.051	96.81 %
45.082	000000	DUA1	44.570	0.512	98.86 %
32.367	DD9RV54	DUA0	31.988	0.379	98.82 %
27.035	DD9JB92	DUA0	25.906	1.129	95.82 %
25.934	000000	DUA0	25.410	0.523	97.98 %
25.852	VMS4	DUA0	25.336	0.516	98.00 %
22.277	SANMN	DUA1	22.234	0.043	99.80 %
20.074	DD9JK54	DUA0	20.039	0.035	99.82 %
12.891	OPERATOR	DUA0	12.668	0.223	98.27 %
8.871	CLASS1	DUA0	8.813	0.059	99.33 %
6.586	OWDFIN	DUA1	6.191	0.395	94.00 %
3.586	DD9GW20	DUA0	3.027	0.559	84.42 %
2.965	SYSE	DUA0	2.938	0.027	99.07 %
2.508	KERMIT	DUA1	2.398	0.109	95.63 %
2.133	PRODUCTS	DUA1	2.070	0.063	97.06 %
1.430	CD9JE15	DUA0	1.172	0.258	81.96 %
1.230	DIABSPKER	DUA0	1.203	0.027	97.77 %
1.219	MD1MP41	DUA0	0.957	0.262	78.52 %
1.184	DD9EC17	DUA0	0.930	0.254	78.54 %
1.172	DSJJZ11	DUA0	0.910	0.262	77.66 %
1.148	DSJDE11	DUA0	0.895	0.254	77.89 %
1.066	PRODUCTS	DUA0	1.035	0.031	97.06 %
0.680	REGHIJBIONS	DUA0	0.660	0.020	97.12 %
0.563	MA9KM38	DUA0	0.273	0.289	48.61 %
0.539	DSJGD11	DUA0	0.254	0.285	47.10 %
0.527	CS9RG99	DUA0	0.406	0.121	77.03 %
0.516	DSJPR11	DUA0	0.250	0.266	48.48 %
0.492	DSJPK11	DUA0	0.227	0.266	46.03 %
0.492	DSJTM11	DUA0	0.227	0.266	46.03 %
0.492	STXDEMO0	DUA0	0.230	0.262	46.82 %
0.410	JVOLM	DUA1	0.352	0.059	85.71 %
0.387	AHZEBEL	DUA1	0.324	0.063	83.83 %

Figure 3-308. Daily Directory Top Level Ranking - Allocated

Daily Directory Top Level Ranking - Unused

INQUIRY: DEXLDB RUN DATE: 24JUL08

System Identifier = VMS2 Year = 08 Month = 7 Day = 9

Megabyte Hours Unused	Directory Top Level	Disk Device Name	Megabyte Hours Allocated	Megabyte Hours Used	Megabyte Hours Pct Used
108.883	SYS0	DUA0	2847.664	2738.781	96.17 %
7.289	ALLIN1	DUA0	227.379	220.090	96.79 %
3.410	JQINJQ	DUA1	249.492	246.082	98.63 %
2.332	PATLM	DUA1	141.949	139.617	98.35 %
2.051	METANET	DUA0	64.313	62.262	96.81 %
1.727	000000	DUA2	88.828	87.102	98.05 %
1.129	DD9JB92	DUA0	27.035	25.906	95.82 %
0.996	FLEX	DUA1	502.195	501.199	99.80 %
0.559	DD9GW20	DUA0	3.586	3.027	84.42 %
0.523	000000	DUA0	25.934	25.410	97.98 %
0.516	VMS4	DUA0	25.852	25.336	98.00 %
0.512	000000	DUA1	45.082	44.570	98.86 %
0.395	OWDFIN	DUA1	6.586	6.191	94.00 %
0.379	DD9RV54	DUA0	32.367	31.988	98.82 %
0.289	MA9KM38	DUA0	0.563	0.273	48.61 %
0.285	DSJGD11	DUA0	0.539	0.254	47.10 %
0.266	DSJPK11	DUA0	0.492	0.227	46.03 %
0.266	DSJPR11	DUA0	0.516	0.250	48.48 %
0.266	DSJTM11	DUA0	0.492	0.227	46.03 %
0.262	DSJJZ11	DUA0	1.172	0.910	77.66 %
0.262	MD1MP41	DUA0	1.219	0.957	78.52 %
0.262	STXDEMO0	DUA0	0.492	0.230	46.82 %
0.258	CD9JE15	DUA0	1.430	1.172	81.96 %
0.254	DD9EC17	DUA0	1.184	0.930	78.54 %
0.254	DSJDE11	DUA0	1.148	0.895	77.89 %
0.250	PERFORM	DUA2	0.340	0.090	26.43 %
0.223	OPERATOR	DUA0	12.891	12.668	98.27 %
0.195	FALAPTCEI	DUA2	255.785	255.590	99.92 %
0.121	CS9RG99	DUA0	0.527	0.406	77.03 %
0.109	KERMIT	DUA1	2.508	2.398	95.63 %
0.063	AHZEBEL	DUA1	0.387	0.324	83.83 %
0.063	PRODUCTS	DUA1	2.133	2.070	97.06 %
0.059	CLASS1	DUA0	8.871	8.813	99.33 %
0.059	JVOLM	DUA1	0.410	0.352	85.71 %
0.051	BACKUP	DUA2	0.070	0.020	27.77 %
0.043	BACKUP	DUA0	0.082	0.039	47.61 %
0.043	SANMN	DUA1	22.277	22.234	99.80 %
0.035	DD9JK54	DUA0	20.074	20.039	99.82 %
0.035	PRINTERSS	DUA1	0.070	0.035	50.00 %
0.031	PRODUCTS	DUA0	1.066	1.035	97.06 %

Figure 3-309. Daily Directory Top Level Ranking - Unused

Weekly Disk Device Free Space

INQUIRY: DEXPWF RUN DATE: 24JUL08

SYSID=VMS2 Year=08

BAR CHART OF DKDBLKSF

```
DEXDEVIC   WEEK   Week                                                  DKDBLKSF
                  |
DUA0         22   |****                                                    75048
             23   |***                                                     64785
             24   |*                                                       27297
             25   |*                                                       20556
             26   |***                                                     55581
             27   |***                                                     54792
                  |
DUA1         22   |**********************************                     673278
             23   |**********************************                     672450
             24   |*********************************                      666801
             25   |*********************************                      658614
             26   |**********************************                     650034
             27   |********************************                       648738
                  |
DUA2         22   |**************************************************     1014696
             23   |***************************************************    1012128
             24   |**************************************************     1003698
             25   |**************************************************     1003698
             26   |**************************************************     1003698
             27   |**************************************************     1003698
                  |
                  ----------+---------+---------+---------+---------+-
                         200000    400000    600000    800000    1000000

                                   Blocks Free
```

Figure 3-310. Weekly Disk Device Free Space

```
+-----------------------------------------------------------------------------------------------------------------------------------+
|                                                Weekly Disk Device Space Utilization                                               |
|                                                                                                                                   |
| INQUIRY:  DEXPWG                                                                                                RUN DATE:  24JUL08|
|                                                                                                                                   |
|                                                        SYSID=VMS2 Year=08                                                         |
|                                                                                                                                   |
|                                                       BAR CHART OF PCTUSED                                                        |
|                                                                                                                                   |
| DEXDEVIC   WEEK   Week                                                                                                   PCTUSED |
|                   |                                                                                                               |
| DUA0         22   |*******************************************************************************************          91.57 % |
|              23   |*********************************************************************************************        92.72 % |
|              24   |*************************************************************************************************    96.93 % |
|              25   |***************************************************************************************************  97.69 % |
|              26   |*********************************************************************************************        93.76 % |
|              27   |*********************************************************************************************        93.85 % |
|                   |                                                                                                               |
| DUA1         22   |************************                                                                                  24.44 % |
|              23   |*************************                                                                                 24.53 % |
|              24   |*************************                                                                                 25.16 % |
|              25   |***************************                                                                               26.08 % |
|              26   |****************************                                                                              27.05 % |
|              27   |****************************                                                                              27.19 % |
|                   |                                                                                                               |
| DUA2         22   |**********                                                                                                10.45 % |
|              23   |***********                                                                                               10.68 % |
|              24   |***********                                                                                               11.42 % |
|              25   |***********                                                                                               11.42 % |
|              26   |***********                                                                                               11.42 % |
|              27   |***********                                                                                               11.42 % |
|                   |                                                                                                               |
|                   ----------+---------+---------+---------+---------+---------+---------+---------+---------+--------                |
|                          10.00 %   20.00 %   30.00 %   40.00 %   50.00 %   60.00 %   70.00 %   80.00 %   90.00 %                  |
|                                                                                                                                   |
|                                                        Percent Space Used                                                         |
|                                                                                                                                   |
|                                                                                                                                   |
|                                   Figure 3-311.   Weekly Disk Device Space Utilization                                            |
|                                                                                                                                   |
|                                                                                                                                   |
|                                                                                                                                   
```

Monthly Disk Usage Summary

INQUIRY: DEXLMA RUN DATE: 24JUL08

System Identifier = VMS2 Year = 08 Month = 6

DEXACT1	DEXACT2	DEXACT3	Megabyte Hours Allocated	Megabyte Hours Used	Megabyte Hours Unused	Megabyte Hours Pct Used
SAL	1	ALLIN1	20099.284	19461.864	637.419	96.82 %
SAL	1	BACKUP	8.298	3.803	4.495	45.83 %
SAL	1	DBDBD	4.149	4.149	0.000	100.00 %
SAL	1	DIABSPKER	108.905	106.485	2.420	97.77 %
SAL	1	FALAPTCEI	21512.979	21495.880	17.099	99.92 %
SAL	1	JFQ	21.781	20.744	1.037	95.23 %
SAL	1	JQINJQ	1.037	1.037	0.000	100.00 %
SAL	1	KAKMTEYE	1.037	0.346	0.691	33.33 %
SAL	1	KCOOK	1.037	0.346	0.691	33.33 %
SAL	1	KERMIT	221.959	212.279	9.680	95.63 %
SAL	1	CA	7.260	4.495	2.766	61.90 %
SAL	1	MENUS	11.409	10.718	0.691	93.93 %
SAL	1	METANET	8278.707	7980.154	298.553	96.39 %
SAL	1	OPERATOR	66.380	48.402	17.978	72.91 %
SAL	1	PATLM	1517.413	1512.228	5.186	99.65 %
SAL	1	PERFORM	11.409	2.074	9.335	18.18 %
SAL	1	PRINTERSS	6.223	3.112	3.112	50.00 %
SAL	1	PRODUCTS	283.154	274.856	8.298	97.06 %
SAL	1	REGHIJBIONS	60.157	58.429	1.729	97.12 %
SAL	1	SYSE	239.592	236.480	3.112	98.70 %
SAL	1	SYSF	2.074	0.691	1.383	33.33 %
SAL	1	SYS0	250453.552	240947.569	9505.983	96.20 %
SAL	1	TEMPUSER	1.037	0.346	0.691	33.33 %
SAL	1	VMS4	2288.048	2242.411	45.636	98.00 %
SAL	1	000000	6202.856	6173.123	29.733	99.52 %
			311409.739	300802.021	10607.718	
SAL	10	PATLM	10871.084	10670.481	200.602	98.15 %
SAL	10	PERFORM	18.669	5.877	12.792	31.48 %
SAL	10	000000	7655.522	7653.448	2.074	99.97 %
			18545.275	18329.806	215.469	
SAL	35	FVVT	3.112	3.112	0.000	100.00 %
SAL	35	000000	1.037	0.346	0.691	33.33 %
			4.149	3.457	0.691	
SAL	56	SYS0	2774.491	2771.726	2.766	99.90 %

Figure 3-312. Monthly Disk Usage Summary

Monthly Disk Usage Ranking - Allocated

INQUIRY: DEXLMB RUN DATE: 24JUL08

System Identifier = VMS2 Year = 08 Month = 6

DEXACT1	DEXACT2	DEXACT3	Megabyte Hours Allocated	Megabyte Hours Used	Megabyte Hours Unused	Megabyte Hours Pct Used
SAL	1	SYS0	250453.552	240947.569	9505.983	96.20 %
		FALAPTCEI	21512.979	21495.880	17.099	99.92 %
		ALLIN1	20099.284	19461.864	637.419	96.82 %
		METANET	8278.707	7980.154	298.553	96.39 %
		000000	6202.856	6173.123	29.733	99.52 %
		VMS4	2288.048	2242.411	45.636	98.00 %
		PATLM	1517.413	1512.228	5.186	99.65 %
		PRODUCTS	283.154	274.856	8.298	97.06 %
		SYSE	239.592	236.480	3.112	98.70 %
		KERMIT	221.959	212.279	9.680	95.63 %
SAL	10	PATLM	10871.084	10670.481	200.602	98.15 %
		000000	7655.522	7653.448	2.074	99.97 %
		PERFORM	18.669	5.877	12.792	31.48 %
SAL	35	FVVT	3.112	3.112	0.000	100.00 %
		000000	1.037	0.346	0.691	33.33 %
SAL	56	SYS0	2774.491	2771.726	2.766	99.90 %
SAL	66	SYS0	1.037	0.346	0.691	33.33 %
SAL	100	FLEX	44426.298	44339.670	86.629	99.80 %
		SWIRLLO	14.521	13.138	1.383	90.47 %
		000000	5.186	2.730	2.456	52.64 %
SAL	200	JQINJQ	14392.070	14122.767	269.303	98.12 %
		SANMN	1971.704	1967.901	3.803	99.80 %
		DD9JB92	1576.945	1504.140	72.805	95.38 %
		DD9RV54	1450.728	1421.018	29.711	97.95 %
		DD9JK54	1145.506	1143.509	1.997	99.82 %
		CLASS1	785.155	779.969	5.186	99.33 %
		DD9GW20	270.686	208.081	62.605	76.87 %
		CD9JE15	126.538	103.719	22.818	81.96 %
		DD9EC17	104.757	82.284	22.473	78.54 %
		MD1MP41	67.299	43.922	23.377	65.26 %
SAL	201	CS9RG99	46.674	35.956	10.718	77.03 %
		000000	1.037	0.346	0.691	33.33 %
SAL	210	000000	1.037	0.346	0.691	33.33 %

Figure 3-313. Monthly Disk Usage Ranking - Allocated

Monthly Disk Usage Ranking - Unused

INQUIRY: DEXLMB RUN DATE: 24JUL08

System Identifier = VMS2 Year = 08 Month = 6

DEXACT1	DEXACT2	DEXACT3	Megabyte Hours Unused	Megabyte Hours Allocated	Megabyte Hours Used	Megabyte Hours Pct Used
SAL	1	SYS0	9505.983	250453.552	240947.569	96.20 %
		ALLIN1	637.419	20099.284	19461.864	96.82 %
		METANET	298.553	8278.707	7980.154	96.39 %
		VMS4	45.636	2288.048	2242.411	98.00 %
		000000	29.733	6202.856	6173.123	99.52 %
		OPERATOR	17.978	66.380	48.402	72.91 %
		FALAPTCEI	17.099	21512.979	21495.880	99.92 %
		KERMIT	9.680	221.959	212.279	95.63 %
		PERFORM	9.335	11.409	2.074	18.18 %
		PRODUCTS	8.298	283.154	274.856	97.06 %
SAL	10	PATLM	200.602	10871.084	10670.481	98.15 %
		PERFORM	12.792	18.669	5.877	31.48 %
		000000	2.074	7655.522	7653.448	99.97 %
SAL	35	000000	0.691	1.037	0.346	33.33 %
		FVVT	0.000	3.112	3.112	100.00 %
SAL	56	SYS0	2.766	2774.491	2771.726	99.90 %
SAL	66	SYS0	0.691	1.037	0.346	33.33 %
SAL	100	FLEX	86.629	44426.298	44339.670	99.80 %
		000000	2.456	5.186	2.730	52.64 %
		SWIRLLO	1.383	14.521	13.138	90.47 %
SAL	200	JQINJQ	269.303	14392.070	14122.767	98.12 %
		DD9JB92	72.805	1576.945	1504.140	95.38 %
		DD9GW20	62.605	270.686	208.081	76.87 %
		DD9RV54	29.711	1450.728	1421.018	97.95 %
		MD1MP41	23.377	67.299	43.922	65.26 %
		CD9JE15	22.818	126.538	103.719	81.96 %
		DD9EC17	22.473	104.757	82.284	78.54 %
		000000	22.473	33.190	10.718	32.29 %
		CLASS1	5.186	785.155	779.969	99.33 %
		SANMN	3.803	1971.704	1967.901	99.80 %
SAL	201	CS9RG99	10.718	46.674	35.956	77.03 %
		000000	0.691	1.037	0.346	33.33 %
SAL	210	000000	0.691	1.037	0.346	33.33 %

Figure 3-314. Monthly Disk Usage Ranking - Unused

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 4

EXCEPTIONS

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 4

EXCEPTIONS

CHAPTER 4. EXCEPTIONS

This chapter is not yet available.

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 5

FILES

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 5

FILES

Chapter 5. FILES

The CA MICS VAX/VMS Analyzer stores its data in the VAX/VMS Accounting (DEA), VAX/VMS Monitor (DEM), and the VAX/VMS System Usage (DES) information areas. The table in Figure 5-1 lists the product's files and the time-spans that are activated for each file. For each file in the information area, the following information is provided:

XDWMYT - Defines the time-spans in which the file is supported.

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

File - The SAS name used to access this file.

File Name - The descriptive label for the file.

VAX/VMS Accounting Information Area			Date Generated: Thu, Jul 18, 2019
XDWMYT	File	File Name	
X..MY.	DEA_LF	VMS ACCOUNTING LOGIN FAILED FILE	
X..M..	DEAIMG	VMS ACCOUNTING IMAGE TERMINATION FILE	
X.....	DEAINT	VMS ACCOUNTING INITIALIZATION FILE	
X..MY.	DEAPRC	VMS ACCOUNTING PROCESS TERMINATION FILE	
X..MY.	DEAPRQ	VMS ACCOUNTING PRINT QUEUED FILE	

VAX/VMS Monitor Information Area			Date Generated: Thu, Jul 18, 2019
XDWMYT	File	File Name	
XDWM..	DEMDSK	VMS MONITOR DISK FILE	
XDWM..	DEMMOD	VMS MONITOR MODES FILE	
XDWM..	DEMPRO	VMS MONITOR PROCESS FILE	
XDWM..	DEMRMS	VMS MONITOR RMS FILE	
XDWM..	DEMSCS	VMS MONITOR SCS FILE	
XDWM..	DEMSPR	VMS SYSTEM PROFILE FILE	

VAX/VMS System Usage Information Area			Date Generated: Thu, Jul 18, 2019
XDWMYT	File	File Name	
XDWM..	DESDKD	VMS DISK DEVICE FILE	
XDWM..	DESDKQ	VMS DISK QUOTA FILE	
XD.MY.	DESDKU	VMS DISK USAGE FILE	
XDWMY.	DESPRX	VMS PROCESS ACTIVITY FILE	
XDWM..	DESSYU	VMS SYSTEM STATUS FILE	

Figure 5-1. VAX/VMS Information Area Files

This chapter contains the following sections:

1 - Data Element Naming Conventions
2 - VAX/VMS Accounting Information Area Files
3 - VAX/VMS Monitor Information Area Files
4 - VAX/VMS System Usage Information Area Files

5.1 - Data Element Naming Conventions

CA MICS data elements follow naming conventions that depend on whether they are standard or common data elements. Standard data elements use the first three characters of their name to identify the file in which they are defined. The following chart lists the file in which they are contained and the three-character prefix with which the standard data element names begin.

File Name	File	Names Begin With
VMS ACCOUNTING Process Termination File	DEAPRC	PRC
VMS ACCOUNTING Image Termination File	DEAIMG	IMG
VMS ACCOUNTING LOGIN Failed File	DEA_LF	PRC
VMS ACCOUNTING Print Queued File	DEAPRQ	PRQ
VMS ACCOUNTING Initialization File	DEAINT	INT
VMS MONITOR Process File	DEMPRO	PRO
VMS MONITOR Modes File	DEMMOD	MOD
VMS MONITOR Disk File	DEMDSK	DSK
VMS MONITOR SCS File	DEMSCS	SCS
VMS MONITOR RMS File	DEMRMS	RMS
VMS System Profile File	DEMSPR	SPR
VMS Disk Device File	DESDKD	DKD
VMS Disk Quota File (DESDKQ)	DESDKQ	DKQ
VMS Disk Usage File (DESDKU)	DESDKU	DKU
VMS Process Activity File (DESPRX)	DESPRX	PRX
VMS System Status File (DESSYU)	DESSYU	SYU

Common data elements do not use a data element prefix. They have a common definition across database information areas or across files within an information area. You will find common data elements listed in the Sequence/Summary Data Elements and Common Data Elements sections of the Data Elements Lists that accompany the following file descriptions.

5.2 - VAX/VMS Accounting Information Area Files

This section identifies each file in the VAX/VMS Accounting information area (DEA), the file's organization, the data elements contained in the file, and the time-spans in which the data elements are supported.

Data elements are described in Appendix B, Data Dictionary.

This section describes the following files:

1 - VMS ACCOUNTING Process Termination File (DEAPRC)
2 - VMS ACCOUNTING Image Termination File (DEAIMG)
3 - VMS ACCOUNTING LOGIN Failed File (DEA_LF)
4 - VMS ACCOUNTING Print Queued File (DEAPRQ)
5 - VMS ACCOUNTING Initialization File (DEAINT)

5.2.1 - VMS ACCOUNTING Process Termination File (DEAPRC)

The VMS ACCOUNTING Process Termination file maintains information on the processes terminated, including starting time, CPU time, page faults, fault I/O, direct and buffered I/O, maximum working set size, and volumes mounted. It is derived from the VMS ACCOUNTING type 1 record.

The following sections describe the file's organization and list the data elements maintained.

1 - DEAPRC File Organization
2 - DEAPRC Data Elements List
3 - DEAPRC Usage Considerations
4 - DEAPRC Retrieval Examples

5.2.1.1 - DEAPRC File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	DAY	HOUR	ENDTS	
DAYS	N/A				
WEEKS	N/A				
MONTHS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	ZONE			
YEARS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	ZONE				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-2. DEAPRC Time-Span Granularity Chart

5.2.1.2 - DEAPRC Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)

Sequence/Summary Data Elements

X.....E	DAY	- Day of Month
X..MY.E	DEXACT1	- DIVISION
X..MY.E	DEXACT2	- UIC GROUP
X..MY.E	DEXACT3	- USER NAME
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..MY.E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year

```
 X..MY.E  YEAR     - Year of Century
 X..MY.E  ZONE     - Time Zone

Common Data Elements

 X.....E  DAYNAME  - Name of Day of Week
 X.....E  DEXACCNT - Account Name
 X.....E  DEXGRP   - User Identification Code Group
 X.....E  DEXJOB   - Jobname
 X.....E  DEXJOBID - Job ID
 X.....E  DEXMBR   - User Identification Code Member
 X.....E  DEXNADDR - Node Address
 X.....E  DEXNODE  - Nodename
 X.....E  DEXOWNID - Owner ID
 X.....E  DEXPID   - Process ID
 X.....E  DEXPRIV1 - Privilege 1 Mask
 X.....E  DEXPRIV2 - Privilege 2 Mask
 X.....E  DEXPRTY  - Priority
 X.....E  DEXQUEUE - Queue Name
 X.....E  DEXRMTID - Remote ID
 X.....E  DEXSTATS - Status
 X.....E  DEXTERM  - Terminal Name
 X.....E  DEXUSER  - Username
 X..MY.E  DURATION - Recording Interval Time
 X..MY.E  ENDTS    - End Time Stamp
 X.....E  ORGSYSID - Originating System Identification
 X..MY.E  STARTTS  - Start Time Stamp

Retained Data Elements

 X.....E  PRCTYPE  - Process Type

Accumulated Data Elements

 X..MY.E  PRCBUFIO - Buffered I/Os
 X..MY.E  PRCCOST  - Processing Charges
 X..MY.E  PRCCPUNI - Number of Instructions Executed
 X..MY.E  PRCCPUTM - Total Process CPU Time
 X..MY.E  PRCDIRIO - Direct I/Os
 X..MY.   PRCFLTIO - Page Fault I/Os
 X..MY.E  PRCIMAGE - Image Count
 X..MY.E  PRCPAGFL - Page File Peak Usage
 X..MY.   PRCPAGFT - Page Faults
 X..MY.E  PRCPRCNT - Process Count
 X..MY.E  PRCVOLS  - Volumes Mounted

Maximum Data Elements

 X..MY.   PRCWKSPK - Working Set Peak
```

5.2.1.3 - DEAPRC Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   o HOUR should not be used in MONTHS and YEARS.
   o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   o WEEK should not be used in MONTHS or YEARS.
   o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL time-span from when they are used in the DAYS, WEEKS, MONTHS, and YEARS time-spans. The ENDTS and STARTTS, when appearing in the higher time-spans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the time-span and ENDTS indicates the end of the time-span.

3. The following data elements only have meaning when using the DEAPRCnn file in the DETAIL time-span, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DEAPRCnn file in the DETAIL time-span.

```
 DEXPID    -  Process ID
 DEXOWNID  -  Owner
 DEXPRIV1  -  Privilege
 DEXPRIV2  -  Privilege
 DEXMBR    -  Member
 DEXGRP    -  Group
 DEXJOBID  -  Job ID
 DEXPRTY   -  Priority
 DEXUSER   -  Username
 DEXNODE   -  Nodename
 DEXACCNT  -  Account
 DEXJOB    -  Jobname
 DEXTERM   -  Terminal
 DEXNADDR  -  Node Address
 DEXQUEUE  -  Queue
 DEXRMTID  -  Remote ID
 DEXTYPE   -  Process Type
 DEXSTATS  -  Status
```

5.2.1.4 - DEAPRC Retrieval Examples

This section presents typical DEAPRC retrieval examples.

1. Print a list of all users and all jobnames for node DEV1 yesterday.

```
DATA;
SET &pDEMX..DEAPRC01;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
PROC PRINT;
 VAR DEXUSER DEXJOB;
```

2. Print the total CPU time used by user FRED on node DEV1 yesterday.

```
DATA;
SET &pDEMX..DEAPRC01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
IF DEXUSER='FRED';
RETAIN CPUTM 0;
FORMAT CPUTM TIME.;
CPUTM+PRCCPUTM;
IF EOF THEN DO;
  PUT PRCCPUTM;
  END;
```

5.2.2 - VMS ACCOUNTING Image Termination File (DEAIMG)

The VMS Accounting Image Termination file maintains information on the images terminated, including starting time, CPU time, page faults, fault I/O, direct and buffered I/O, maximum working set size, and volumes mounted. It is derived from the VMS ACCOUNTING type 3 record.

The following sections describe the file's organization and list the data elements maintained.

1 - DEAIMG File Organization
2 - DEAIMG Data Elements List
3 - DEAIMG Usage Considerations
4 - DEAIMG Retrieval Examples

5.2.2.1 - DEAIMG File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	DAY	HOUR	ENDTS	
DAYS	N/A				
WEEKS	N/A				
MONTHS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	ZONE			
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-3. DEAIMG Time-Span Granularity Chart

5.2.2.2 - DEAIMG Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)

Sequence/Summary Data Elements

X.....E	DAY	- Day of Month
X..M..E	DEXACT1	- DIVISION
X..M..E	DEXACT2	- UIC GROUP
X..M..E	DEXACT3	- USER NAME
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year

```
 X..M..E  YEAR     - Year of Century
 X..M..E  ZONE     - Time Zone

Common Data Elements

 X.....E  DAYNAME  - Name of Day of Week
 X.....E  DEXACCNT - Account Name
 X.....E  DEXGRP   - User Identification Code Group
 X.....E  DEXJOB   - Jobname
 X.....E  DEXJOBID - Job ID
 X.....E  DEXMBR   - User Identification Code Member
 X.....E  DEXNADDR - Node Address
 X.....E  DEXNODE  - Nodename
 X.....E  DEXOWNID - Owner ID
 X.....E  DEXPID   - Process ID
 X.....E  DEXPRIV1 - Privilege 1 Mask
 X.....E  DEXPRIV2 - Privilege 2 Mask
 X.....E  DEXPRTY  - Priority
 X.....E  DEXQUEUE - Queue Name
 X.....E  DEXRMTID - Remote ID
 X.....E  DEXSTATS - Status
 X.....E  DEXTERM  - Terminal Name
 X.....E  DEXUSER  - Username
 X..M..E  DURATION - Recording Interval Time
 X..M..E  ENDTS    - End Time Stamp
 X.....E  ORGSYSID - Originating System Identification
 X..M..E  STARTTS  - Start Time Stamp

Retained Data Elements

 X.....   IMGIMAGE - Image Sequence Number
 X.....E  IMGNAME  - Image Name
 X.....E  IMGTYPE  - Process Type

Accumulated Data Elements

 X..M..E  IMGBUFIO - Buffered I/Os
 X..M..E  IMGCOST  - Processing Charges
 X..M..E  IMGCPUNI - Number of Instructions Executed
 X..M..E  IMGCPUTM - Total Image CPU Time
 X..M..E  IMGDIRIO - Direct I/Os
 X..M..   IMGFLTIO - Page Fault I/Os
 X..M..E  IMGIMCNT - Image Count
 X..M..   IMGPAGFL - Page File Peak Usage
 X..M..   IMGPAGFT - Page Faults
 X..M..E  IMGVOLS  - Volumes Mounted

Maximum Data Elements

 X..M..   IMGWKSPK - Working Set Peak
```

5.2.2.3 - DEAIMG Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   o HOUR should not be used in MONTHS and YEARS.
   o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   o WEEK should not be used in MONTHS or YEARS.
   o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS time-spans. The ENDTS and STARTTS, when appearing in the higher time-spans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the time-span and ENDTS indicates the end of the time-span.

3. The following data elements only have meaning when using the DEAIMGnn file in the DETAIL timespan, in that they loose significance once summarization has been performed. These data elements should only be referenced when using the DEAIMGnn file in the DETAIL timespan.

```
 DEXPID    -  Process ID
 DEXOWNID  -  Owner
 DEXPRIV1  -  Privilege
 DEXPRIV2  -  Privilege
 DEXMBR    -  Member
 DEXGRP    -  Group
 DEXJOBID  -  Job ID
 DEXPRTY   -  Priority
 DEXUSER   -  Username
 DEXNODE   -  Nodename
 DEXACCNT  -  Account
 DEXJOB    -  Jobname
 DEXTERM   -  Terminal
 DEXNADDR  -  Node Address
 DEXQUEUE  -  Queue
 DEXRMTID  -  Remote ID
 DEXTYPE   -  Process Type
 DEXSTATS  -  Status
 IMGNAME   -  Image Name
```

5.2.2.4 - DEAIMG Retrieval Examples

This section presents typical DEAIMG retrieval examples.

1. Print a list of all users and all images for node DEV1 yesterday.

```
DATA;
SET &pDEMX..DEAIMG01;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
PROC PRINT;
 VAR DEXUSER IMGNAME;
```

2. Print the total volumes mounted on node DEV1 yesterday.

```
DATA;
SET &pDEMX..DEAIMG01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
RETAIN VOLS 0;
VOLS+IMGVOLS;
IF EOF THEN DO;
  PUT VOLS;
  END;
```

5.2.3 - VMS ACCOUNTING LOGIN Failed File (DEA_LF)

The VMS ACCOUNTING LOGIN Failed file maintains information on unsuccessful attempts to gain access to the system. It is derived from the VMS ACCOUNTING type 7 record.

The following sections describe the file's organization and list the data elements maintained.

1 - DEA_LF File Organization
2 - DEA_LF Data Elements List
3 - DEA_LF Usage Considerations
4 - DEA_LF Retrieval Example

5.2.3.1 - DEA_LF File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	DAY	HOUR	ENDTS	
DAYS	N/A				
WEEKS	N/A				
MONTHS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	ZONE			
YEARS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	ZONE				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-4. DEA_LF Time-Span Granularity Chart

5.2.3.2 - DEA_LF Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
X.....E	DAY	- Day of Month
X..MY.E	DEXACT1	- DIVISION
X..MY.E	DEXACT2	- UIC GROUP
X..MY.E	DEXACT3	- USER NAME
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..MY.E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year

```
 X..MY.E  YEAR     - Year of Century
 X..MY.E  ZONE     - Time Zone

Common Data Elements

 X.....E  DAYNAME  - Name of Day of Week
 X.....E  DEXACCNT - Account Name
 X.....E  DEXGRP   - User Identification Code Group
 X.....E  DEXJOB   - Jobname
 X.....E  DEXJOBID - Job ID
 X.....E  DEXMBR   - User Identification Code Member
 X.....E  DEXNADDR - Node Address
 X.....E  DEXNODE  - Nodename
 X.....E  DEXOWNID - Owner ID
 X.....E  DEXPID   - Process ID
 X.....E  DEXPRIV1 - Privilege 1 Mask
 X.....E  DEXPRIV2 - Privilege 2 Mask
 X.....E  DEXPRTY  - Priority
 X.....E  DEXQUEUE - Queue Name
 X.....E  DEXRMTID - Remote ID
 X.....E  DEXSTATS - Status
 X.....E  DEXTERM  - Terminal Name
 X.....E  DEXUSER  - Username
 X..MY.E  DURATION - Recording Interval Time
 X..MY.E  ENDTS    - End Time Stamp
 X.....E  ORGSYSID - Originating System Identification
 X..MY.E  STARTTS  - Start Time Stamp

Accumulated Data Elements

 X..MY.E  PRCBUFIO - Buffered I/Os
 X..MY.E  PRCCOST  - Processing Charges
 X..MY.E  PRCCPUNI - Number of Instructions Executed
 X..MY.E  PRCCPUTM - Total CPU Time
 X..MY.E  PRCDIRIO - Direct I/Os
 X..MY.   PRCFLTIO - Page Fault I/Os
 X..MY.   PRCIMAGE - Image Count
 X..MY.   PRCPAGFL - Page File Peak Usage
 X..MY.   PRCPAGFT - Page Faults
 X..MY.E  PRCVOLS  - Volumes Mounted

Maximum Data Elements

 X..MY.   PRCWKSPK - Working Set Peak
```

5.2.3.3 - DEA_LF Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   - o HOUR should not be used in MONTHS and YEARS.
   - o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   - o WEEK should not be used in MONTHS or YEARS.
   - o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DEA_LFnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DEA_LFnn file in the DETAIL timespan.

```
DEXPID    -  Process ID
DEXOWNID  -  Owner
DEXPRIV1  -  Privilege
DEXPRIV2  -  Privilege
DEXMBR    -  Member
DEXGRP    -  Group
DEXJOBID  -  Job ID
DEXPRTY   -  Priority
DEXUSER   -  Username
DEXNODE   -  Nodename
DEXACCNT  -  Account
DEXJOB    -  Jobname
DEXTERM   -  Terminal
DEXNADDR  -  Node Address
DEXQUEUE  -  Queue
DEXRMTID  -  Remote ID
DEXSTATS  -  Status
```

5.2.3.4 - DEA_LF Retrieval Examples

This section presents a typical DEA_LF retrieval example.

1. Print a list of all terminals that had LOGIN failures for node DEV1 yesterday.

```
DATA;
SET &pDEMX..DEA_LF01;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
PROC PRINT;
 VAR DEXTERM;
```

5.2.4 - VMS ACCOUNTING Print Queued File (DEAPRQ)

The VMS ACCOUNTING Print Queued file maintains information on printing, including start time, queue time, pages, and I/O count. It is derived from the VMS ACCOUNTING type 8 record.

The following sections describe the file's organization and list the data elements maintained.

1 - DEAPRQ File Organization
2 - DEAPRQ Data Elements List
3 - DEAPRQ Usage Considerations
4 - DEAPRQ Retrieval Examples

5.2.4.1 - DEAPRQ File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	DAY	HOUR	ENDTS	
DAYS	N/A				
WEEKS	N/A				
MONTHS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	ZONE			
YEARS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	ZONE				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-5. DEAPRQ Time-Span Granularity Chart

5.2.4.2 - DEAPRQ Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
X.....E	DAY	- Day of Month
X..MY.E	DEXACT1	- DIVISION
X..MY.E	DEXACT2	- UIC GROUP
X..MY.E	DEXACT3	- USER NAME
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..MY.E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year

```
 X..MY.E  YEAR     - Year of Century
 X..MY.E  ZONE     - Time Zone

Common Data Elements

 X.....E  DAYNAME  - Name of Day of Week
 X.....E  DEXACCNT - Account Name
 X.....E  DEXGRP   - User Identification Code Group
 X.....E  DEXJOB   - Jobname
 X.....E  DEXJOBID - Job ID
 X.....E  DEXMBR   - User Identification Code Member
 X.....E  DEXNADDR - Node Address
 X.....E  DEXNODE  - Nodename
 X.....E  DEXOWNID - Owner ID
 X.....E  DEXPID   - Process ID
 X.....E  DEXPRIV1 - Privilege 1 Mask
 X.....E  DEXPRIV2 - Privilege 2 Mask
 X.....E  DEXPRTY  - Priority
 X.....E  DEXQUEUE - Queue Name
 X.....E  DEXRMTID - Remote ID
 X.....E  DEXSTATS - Status
 X.....E  DEXTERM  - Terminal Name
 X.....E  DEXUSER  - Username
 X..MY.E  DURATION - Recording Interval Time
 X..MY.E  ENDTS    - End Time Stamp
 X.....E  ORGSYSID - Originating System Identification
 X..MY.E  STARTTS  - Start Time Stamp

Accumulated Data Elements

 X..MY.E  PRQCOST  - Processing Charges
 X..MY.E  PRQGETS  - GET Count
 X..MY.E  PRQPAGES - Print Pages
 X..MY.E  PRQQIOS  - QIO Count
 X..MY.E  PRQQUETM - Queue Time
 X..MY.E  PRQSYMTM - Symbiont Time

Minimum Data Elements

 X..MY.E  PRQQUETS - Queue Time Stamp
```

5.2.4.3 - DEAPRQ Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   - o HOUR should not be used in MONTHS and YEARS.
   - o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   - o WEEK should not be used in MONTHS or YEARS.
   - o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DEAPRQnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DEAPRQnn file in the DETAIL timespan.

```
DEXPID    -  Process ID
DEXOWNID  -  Owner
DEXPRIV1  -  Privilege
DEXPRIV2  -  Privilege
DEXMBR    -  Member
DEXGRP    -  Group
DEXJOBID  -  Job ID
DEXPRTY   -  Priority
DEXUSER   -  Username
DEXNODE   -  Nodename
DEXACCNT  -  Account
DEXJOB    -  Jobname
DEXTERM   -  Terminal
DEXNADDR  -  Node Address
DEXQUEUE  -  Queue
DEXRMTID  -  Remote ID
DEXSTATS  -  Status
```

5.2.4.4 - DEAPRQ Retrieval Examples

This section presents typical DEAPRQ retrieval examples.

1. Print a list of all users who used printers yesterday.

```
DATA;
SET &pDEMX..DEAPRQ01;
IF DATEPART(ENDTS)=TODAY()-1;
PROC FREQ;
 TABLE DEXUSER;
```

2. Print the total number of lines printed on node DEV1 yesterday.

```
DATA;
SET &pDEMX..DEAPRC01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
RETAIN PAGES 0;
PAGES+PRQPAGES;
IF EOF THEN DO;
  PUT PAGES;
  END;
```

5.2.5 - VMS ACCOUNTING Initialization File (DEAINT)

The VMS ACCOUNTING Initialization file maintains information on system initializations. It is derived from the VMS ACCOUNTING type 5 record.

The following sections describe the file's organization and list the data elements maintained.

1 - DEAINT File Organization
2 - DEAINT Data Elements List
3 - DEAINT Usage Considerations
4 - DEAINT Retrieval Example

5.2.5.1 - DEAINT File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID MONTH	DEXACT1 DAY	DEXACT2 HOUR	DEXACT3 ENDTS	YEAR
DAYS	N/A				
WEEKS	N/A				
MONTHS	N/A				
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-6. DEAINT Time-Span Granularity Chart

5.2.5.2 - DEAINT Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
X.....E	DAY	- Day of Month
X.....E	DEXACT1	- DIVISION
X.....E	DEXACT2	- UIC GROUP
X.....E	DEXACT3	- USER NAME
X.....E	HOUR	- Hour of Day
X.....E	MONTH	- Month of Year
X.....E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year

```
 X.....E  YEAR     - Year of Century
 X.....E  ZONE     - Time Zone

Common Data Elements

 X.....E  DAYNAME  - Name of Day of Week
 X.....E  DEXACCNT - Account Name
 X.....E  DEXGRP   - User Identification Code Group
 X.....E  DEXJOB   - Jobname
 X.....E  DEXJOBID - Job ID
 X.....E  DEXMBR   - User Identification Code Member
 X.....E  DEXNADDR - Node Address
 X.....E  DEXNODE  - Nodename
 X.....E  DEXOWNID - Owner ID
 X.....E  DEXPID   - Process ID
 X.....E  DEXPRIV1 - Privilege 1 Mask
 X.....E  DEXPRIV2 - Privilege 2 Mask
 X.....E  DEXPRTY  - Priority
 X.....E  DEXQUEUE - Queue Name
 X.....E  DEXRMTID - Remote ID
 X.....E  DEXSTATS - Status
 X.....E  DEXTERM  - Terminal Name
 X.....E  DEXUSER  - Username
 X.....E  DURATION - Recording Interval Time
 X.....E  ENDTS    - End Time Stamp
 X.....E  ORGSYSID - Originating System Identification
 X.....E  STARTTS  - Start Time Stamp

Accumulated Data Elements

 X.....E  INTBUFIO - Buffered I/Os
 X.....E  INTCPUNI - Number of Instructions Executed
 X.....E  INTCPUTM - CPU Time
 X.....E  INTDIRIO - Direct I/Os
 X.....   INTFLTIO - Page Fault I/Os
 X.....   INTIMAGE - Image Count
 X.....   INTPAGFL - Page File Peak Usage
 X.....   INTPAGFT - Page Faults
 X.....E  INTVOLS  - Volumes Mounted

Maximum Data Elements

 X.....   INTWKSPK - Working Set Peak
```

5.2.5.3 - DEAINT Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   - o HOUR should not be used in MONTHS and YEARS.
   - o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   - o WEEK should not be used in MONTHS or YEARS.
   - o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DEAINTnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DEAINTnn file in the DETAIL timespan.

```
DEXPID    -  Process ID
DEXOWNID  -  Owner
DEXPRIV1  -  Privilege
DEXPRIV2  -  Privilege
DEXMBR    -  Member
DEXGRP    -  Group
DEXJOBID  -  Job ID
DEXPRTY   -  Priority
DEXUSER   -  Username
DEXNODE   -  Nodename
DEXACCNT  -  Account
DEXJOB    -  Jobname
DEXTERM   -  Terminal
DEXNADDR  -  Node Address
DEXQUEUE  -  Queue
DEXRMTID  -  Remote ID
DEXSTATS  -  Status
```

5.2.5.4 - DEAINT Retrieval Example

This section presents a typical DEAINT retrieval example.

1. Print the times all systems were initialized yesterday.

```
DATA;
SET &pDEMX..DEAINT01;
IF DATEPART(ENDTS)=TODAY()-1;
PROC PRINT;
 VAR SYSID STARTTS;
```

5.3 - VAX/VMS Monitor Information Area Files

This section identifies each file in the VAX/VMS Monitor information area (DEM), the file's organization, the data elements contained in the file, and the time-spans in which the data elements are supported.

Data elements are described in Appendix B, Data Dictionary.

This section describes the following files:

1 - VMS MONITOR Process File (DEMPRO)
2 - VMS MONITOR Modes File (DEMMOD)
3 - VMS MONITOR Disk File (DEMDSK)
4 - VMS MONITOR SCS File (DEMSCS)
5 - VMS MONITOR RMS File (DEMRMS)
6 - VMS System Profile File (DEMSPR)

5.3.1 - VMS MONITOR Process File (DEMPRO)

The VMS MONITOR Process file maintains information on the processes executing in the system, including process name, page count, I/Os, and CPU time. It is derived from the VMS MONITOR type 0 record.

The following sections describe the file's organization and list the data elements maintained.

1 - DEMPRO File Organization
2 - DEMPRO Data Elements List
3 - DEMPRO Usage Considerations
4 - DEMPRO Retrieval Examples

5.3.1.1 - DEMPRO File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	PROPID	PROPNAME	YEAR	MONTH
	DAY	HOUR	ENDTS		
DAYS	SYSID	PROPNAME	YEAR	MONTH	DAY
	HOUR				
WEEKS	SYSID	PROPNAME	YEAR	WEEK	ZONE
	HOUR				
MONTHS	SYSID	PROPNAME	YEAR	MONTH	ZONE
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-7. DEMPRO Time-Span Granularity Chart

5.3.1.2 - DEMPRO Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)

Sequence/Summary Data Elements

XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year
X.....E	PROPID	- Process ID
XDWM..E	PROPNAME	- Process Name
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century

```
XDWM..E  ZONE     - Time Zone
```

Common Data Elements

```
XD....E  DAYNAME  - Name of Day of Week
XDWM..   DEXBOOTS - Monitor System Boot Time Stamp
XDWM..E  DEXDUR   - Requested Recording Interval Time
X.....E  DEXINTNO - Reserved Field Incremented by Monitor
X.....E  DEXMONVR - VMS Monitor Version
XDWM..E  DEXTS    - Monitor Time Stamp
XDWM..E  DURATION - Recording Interval Time
XDWM..E  ENDTS    - End Time Stamp
XDWM..E  INTERVLS - Number of Recording Intervals
X.....E  ORGSYSID - Originating System Identification
XDWM..E  STARTTS  - Start Time Stamp
```

Retained Data Elements

```
X.....E  PROEID   - Extended Process ID
X.....   PROEVNTF - Event Flag Wait Mask
X.....E  PROGLBPG - Global Page Count
X.....E  PROGRP   - User Identification Code Group
X.....E  PROMBR   - User Identification Code Member
X.....E  PROPRCPG - Process Page Count
X.....   PROPRTY  - Priority
X.....E  PROSTATE - State
X.....   PROSTATF - Status Flags
```

Accumulated Data Elements

```
XDWM..E  PROBUFIO - Buffered I/Os
XDWM..E  PROCPUNI - Number of Instructions Executed
XDWM..E  PROCPUTM - CPU Time
XDWM..E  PRODIRIO - Direct I/Os
XDWM..E  PROPGFLT - Page Faults
```

Maximum Data Elements

```
XDWM..E  PROMXGPG - Maximum Global Page Count
XDWM..E  PROMXPPG - Maximum Process Page Count
```

5.3.1.3 - DEMPRO Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   - o HOUR should not be used in MONTHS and YEARS.
   - o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   - o WEEK should not be used in MONTHS or YEARS.
   - o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DEMPROnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DEMPROnn file in the DETAIL timespan.

```
PROEID    -  Extended Process ID
PROPNAME  -  Process Name
PROGLBPG  -  Global Page Count
PROPRCPG  -  Process Page Count
PROSTATF  -  Status Flags
PROEVNTF  -  Event Flag Wait Flag
PROPID    -  Process ID
PROPRTY   -  Priority
PROMBR    -  Member Identification Code
PROGRP    -  User Identification Code
PROSTATE  -  State
```

5.3.1.4 - DEMPRO Retrieval Examples

This section presents typical DEMPRO retrieval examples.

1. Print the total CPU time, direct I/Os, and page faults of all processes of group 4000 and user 300 on node DEV1 yesterday.

```
DATA;
SET &pDEMX..DEMPRO01 END=EOF;
IF SYSID='DEV1';
IF DATEPART(ENDTS)=TODAY()-1;
IF PROGRP=4000 AND PROMBR=300;
RETAIN CPUTM DIO FLTS 0;
CPUTM+PROCPUTM;
DIO+PRODIRIO;
FLTS+PROPGFLT;
IF EOF THEN DO;
  PUT CPUTM DIO FLTS;
  END;
```

2. Print the maximum number of process pages used by all processes yesterday with a group ID larger than 10.

```
DATA;
SET &pDEMX..DEMPRO01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
IF PROGRP GT 10;
RETAIN MAXPGS 0;
MAXPGS=MAX(PROPRCPG,MAXPGS);
IF EOF THEN DO;
  PUT MAXPGS;
  END;
```

5.3.2 - VMS MONITOR Modes File (DEMMOD)

The VMS MONITOR Modes file maintains information describing the time spent in each of the processor modes--interrupt, synchronization, kernel, executive, supervisor, user, compatibility, and idle. It is derived from the VMS MONITOR type 2 record.

This file also maintains CPU time of Vector Consumers, which is derived from the VMS MONITOR type 23 record.

The following sections describe the file's organization and list the data elements maintained.

1 - DEMMOD File Organization
2 - DEMMOD Data Elements List
3 - DEMMOD Usage Considerations
4 - DEMMOD Retrieval Examples

5.3.2.1 - DEMMOD File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	MODCPU	YEAR	MONTH	DAY
	HOUR	ENDTS			
DAYS	SYSID	MODCPU	YEAR	MONTH	DAY
	HOUR				
WEEKS	SYSID	MODCPU	YEAR	WEEK	ZONE
	HOUR				
MONTHS	SYSID	MODCPU	YEAR	MONTH	ZONE
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-8. DEMMOD Time-Span Granularity Chart

5.3.2.2 - DEMMOD Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XDWM..E	MODCPU	- CPU Identifier
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
XDWM..E	ZONE	- Time Zone

Common Data Elements

```
 XD....E  DAYNAME  - Name of Day of Week
 XDWM..   DEXBOOTS - Monitor System Boot Time Stamp
 XDWM..E  DEXDUR   - Requested Recording Interval Time
 X.....E  DEXINTNO - Reserved Field Incremented by Monitor
 X.....E  DEXMONVR - VMS Monitor Version
 XDWM..E  DEXTS    - Monitor Time Stamp
 XDWM..E  DURATION - Recording Interval Time
 XDWM..E  ENDTS    - End Time Stamp
 XDWM..E  INTERVLS - Number of Recording Intervals
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  STARTTS  - Start Time Stamp
```

Accumulated Data Elements

```
 XDWM..E  MODCOMTM - Compatibility Mode Time
 XDWM..   MODCPUNI - Number of Instructions Executed
 XDWM..E  MODEXETM - Executive Mode Time
 XDWM..E  MODIDLTM - Idle Time
 XDWM..E  MODINTTM - Interrupt Stack Time
 XDWM..E  MODKERTM - Kernel Mode Time
 XDWM..E  MODMPSTM - Multi-Processor Sync Time
 XDWM..E  MODSUPTM - Supervisor Mode Time
 XDWM..E  MODUSRTM - User Mode Time
 XDWM..E  MODVECTM - CPU Time of Vector Consumers
```

Derived Data Elements

```
 XDWM..E  MODCPUTM - Total CPU Time
 XDWM..E  MODPCCOM - Compatibility Mode Percent
 XDWM..E  MODPCCPU - Total CPU Percent
 XDWM..E  MODPCEXE - Executive Mode Percent
 XDWM..E  MODPCIDL - Idle Percent
 XDWM..E  MODPCINT - Interrupt Stack Percent
 XDWM..E  MODPCKER - Kernel Mode Percent
 XDWM..E  MODPCMPS - Multi-Processor Sync Percent
 XDWM..E  MODPCSUP - Supervisor Mode Percent
 XDWM..E  MODPCUSR - User Mode Percent
```

5.3.2.3 - DEMMOD Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   - o HOUR should not be used in MONTHS and YEARS.
   - o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   - o WEEK should not be used in MONTHS or YEARS.
   - o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

5.3.2.4 - DEMMOD Retrieval Examples

This section presents typical DEMMOD retrieval examples.

1. Print the ratio of the time spent in supervisor mode to the sum of time in supervisor and user modes for node DEV1 yesterday.

```
DATA;
SET &pDEMX..DEMMOD01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
RETAIN SUP SUPUSR 0;
SUP+MODSUPTM;
SUPUSR=SUPUSR+MODSUPTM+MODUSRTM;
IF EOF THEN DO;
  IF SUPUSR GT 0 THEN RATIO=SUP/SUPUSR;
  PUT SUP SUPUSR RATIO;
  END;
```

2. Print the percentage of idle time for node DEV1 yesterday.

```
DATA;
SET &pDEMX..DEMMOD01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
RETAIN IDLE DUR 0;
IDLE+MODIDLTM;
DUR+DURATION;
IF EOF THEN DO;
  IDLEPCT=IDLE/DUR;
  PUT IDLEPCT;
  END;
```

5.3.3 - VMS MONITOR Disk File (DEMDSK)

The VMS MONITOR Disk file maintains information on VMS disk activity by volume, including number of I/O operations and queue lengths. It is derived from the VMS MONITOR type 12 record.

The following sections describe the file's organization and list the data elements maintained.

1 - DEMDSK File Organization
2 - DEMDSK Data Elements List
3 - DEMDSK Usage Considerations
4 - DEMDSK Retrieval Examples

5.3.3.1 - DEMDSK File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DSKNODE	DSKCTRLR	DSKUNIT	DSKVOL
	YEAR	MONTH	DAY	HOUR	ENDTS
DAYS	SYSID	DSKNODE	DSKCTRLR	DSKUNIT	DSKVOL
	YEAR	MONTH	DAY	HOUR	
WEEKS	SYSID	DSKNODE	DSKCTRLR	DSKUNIT	DSKVOL
	YEAR	WEEK	ZONE	HOUR	
MONTHS	SYSID	DSKNODE	DSKCTRLR	DSKUNIT	DSKVOL
	YEAR	MONTH	ZONE		
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-9. DEMDSK Time-Span Granularity Chart

5.3.3.2 - DEMDSK Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XDWM..E	DSKCTRLR	- Disk Controller
XDWM..E	DSKNODE	- Nodename
XDWM..E	DSKUNIT	- Disk Unit Number
XDWM..E	DSKVOL	- Disk Volume Name
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier

```
 XDW...E  WEEK     - Week of Year
 XDWM..E  YEAR     - Year of Century
 XDWM..E  ZONE     - Time Zone

Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 XDWM..   DEXBOOTS - Monitor System Boot Time Stamp
 XDWM..E  DEXDUR   - Requested Recording Interval Time
 X.....E  DEXINTNO - Reserved Field Incremented by Monitor
 X.....E  DEXMONVR - VMS Monitor Version
 XDWM..E  DEXTS    - Monitor Time Stamp
 XDWM..E  DURATION - Recording Interval Time
 XDWM..E  ENDTS    - End Time Stamp
 XDWM..E  INTERVLS - Number of Recording Intervals
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  STARTTS  - Start Time Stamp

Retained Data Elements

 X.....   DSKALLOC - Allocation Class
 X.....   DSKFLAGS - Flags

Accumulated Data Elements

 XDWM..E  DSKIOOPS - Disk I/O Operations
 XDWM..E  DSKIOQUE - Disk I/O Queue Length Samples Sum

Derived Data Elements

 XDWM..E  DSKAVRES - Disk Average Response Time
 XDWM..E  DSKPSIO  - Disk I/O Operation Rate
```

5.3.3.3 - DEMDSK Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   o HOUR should not be used in MONTHS and YEARS.
   o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   o WEEK should not be used in MONTHS or YEARS.
   o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DEMDSKnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DEMDSKnn file in the DETAIL timespan.

   DSKALLOC - Allocation Class
   DSKFLAGS - Flags

5.3.3.4 - DEMDSK Retrieval Examples

This section presents typical DEMDSK retrieval examples.

1. Print the number of I/O operations per hour for disk 122DRA2: for Zone 1 yesterday.

```
DATA;
SET &pDEMD..DEMDSK01;
IF DATEPART(ENDTS)=TODAY()-1;
IF ZONE=1;
IF DSKVOL='122$DRA2:';
PROC PRINT;
 VAR SYSID MONTH DAY YEAR HOUR DSKIOOPS;
```

2. Plot the average response time for disk 122DRA2: for Zone 1 yesterday.

```
DATA;
SET &pDEMX..DEMDSK01;
IF DATEPART(ENDTS)=TODAY()-1;
IF ZONE=1;
IF DSKVOL='122$DRA2:';
PROC PLOT;
 PLOT DSKAVRES*ENDTS;
```

5.3.4 - VMS MONITOR SCS File (DEMSCS)

The VMS MONITOR SCS file maintains information on the VMS System Communication Services activity, including datagrams sent, received, and discarded; send-datas; messages sent and received; and kilobytes transferred. It is derived from the VMS MONITOR type 15 record.

The following sections describe the file's organization and list the data elements maintained.

1 - DEMSCS File Organization
2 - DEMSCS Data Elements List
3 - DEMSCS Usage Considerations
4 - DEMSCS Retrieval Examples

5.3.4.1 - DEMSCS File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	SCSNODE	YEAR	MONTH	DAY
	HOUR	ENDTS			
DAYS	SYSID	SCSNODE	YEAR	MONTH	DAY
	HOUR				
WEEKS	SYSID	SCSNODE	YEAR	WEEK	ZONE
	HOUR				
MONTHS	SYSID	SCSNODE	YEAR	MONTH	ZONE
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-10. DEMSCS Time-Span Granularity Chart

5.3.4.2 - DEMSCS Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)

Sequence/Summary Data Elements

XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year
XDWM..E	SCSNODE	- Node Identifier
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
XDWM..E	ZONE	- Time Zone

Common Data Elements

```
XD....E  DAYNAME  - Name of Day of Week
XDWM..   DEXBOOTS - Monitor System Boot Time Stamp
XDWM..E  DEXDUR   - Requested Recording Interval Time
X.....E  DEXINTNO - Reserved Field Incremented by Monitor
X.....E  DEXMONVR - VMS Monitor Version
XDWM..E  DEXTS    - Monitor Time Stamp
XDWM..E  DURATION - Recording Interval Time
XDWM..E  ENDTS    - End Time Stamp
XDWM..E  INTERVLS - Number of Recording Intervals
X.....E  ORGSYSID - Originating System Identification
XDWM..E  STARTTS  - Start Time Stamp
```

Accumulated Data Elements

```
XDWM..E  SCSBTKBM - Block Transfer Kilobytes Mapped
XDWM..   SCSBTRD  - Block Transfer Request Datas
XDWM..   SCSBTSND - Block Transfer Send Datas
XDWM..   SCSCQBFR - Connections Queued for Buffer Descript
XDWM..   SCSCQSND - Connections Queued for Send Credit
XDWM..E  SCSDGDIS - Datagrams Discarded
XDWM..E  SCSDGRCV - Datagrams Received
XDWM..E  SCSDGSNT - Datagrams Sent
XDWM..   SCSKBRCV - Kilobytes Received by Request Datas
XDWM..   SCSKBSND - Kilobytes Sent by Send Datas
XDWM..E  SCSMSGRC - Sequenced Messages Received
XDWM..E  SCSMSGSN - Sequenced Messages Sent
```

Derived Data Elements

```
XDWM..E  SCSPSKBT - Block Transfer Kilobytes Mapped Rate
```

5.3.4.3 - DEMSCS Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   - o HOUR should not be used in MONTHS and YEARS.
   - o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   - o WEEK should not be used in MONTHS or YEARS.
   - o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

5.3.4.4 - DEMSCS Retrieval Examples

This section presents typical DEMSCS retrieval examples.

1. Plot the kilobytes sent and received over time for node DEV1 yesterday during prime time (Zone 1).

```
DATA;
SET &pDEMX..DEMSCS01;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
IF ZONE=1;
PROC PLOT;
 PLOT SCSKBSND*ENDTS='S';
 PLOT SCSKBRCV*ENDTS='R';
```

2. Print the sum of kilobytes sent and received yesterday on node DEV1.

```
DATA;
SET &pDEMX..DEMSCS01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
RETAIN KB 0;
KB=KB+SCSKBSND+SCSKBRCV;
IF EOF THEN DO;
  PUT KB;
  END;
```

5.3.5 - VMS MONITOR RMS File (DEMRMS)

The VMS MONITOR RMS file maintains information on the VMS Record Management Services, including, among others, GET bytes, PUT bytes, update bytes, deletes, finds, connects, and disconnects. It is derived from the VMS MONITOR type 20 record.

The following sections describe the file's organization and list the data elements maintained.

1 - DEMRMS File Organization
2 - DEMRMS Data Elements List
3 - DEMRMS Usage Considerations
4 - DEMRMS Retrieval Examples

5.3.5.1 - DEMRMS File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	RMSFILE	YEAR	MONTH	DAY
	HOUR	ENDTS			
DAYS	SYSID	RMSFILE	YEAR	MONTH	DAY
	HOUR				
WEEKS	SYSID	RMSFILE	YEAR	WEEK	ZONE
	HOUR				
MONTHS	SYSID	RMSFILE	YEAR	MONTH	ZONE
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-11. DEMRMS Time-Span Granularity Chart

5.3.5.2 - DEMRMS Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)

Sequence/Summary Data Elements

XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year
XDWM..E	RMSFILE	- File Number
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
XDWM..E	ZONE	- Time Zone

```
Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 XDWM..   DEXBOOTS - Monitor System Boot Time Stamp
 XDWM..E  DEXDUR   - Requested Recording Interval Time
 X.....E  DEXINTNO - Reserved Field Incremented by Monitor
 X.....E  DEXMONVR - VMS Monitor Version
 XDWM..E  DEXTS    - Monitor Time Stamp
 XDWM..E  DURATION - Recording Interval Time
 XDWM..E  ENDTS    - End Time Stamp
 XDWM..E  INTERVLS - Number of Recording Intervals
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  STARTTS  - Start Time Stamp

Retained Data Elements

 X.....E  RMSFILNM - File Name
 X.....E  RMSORG   - Organization

Accumulated Data Elements

 XDWM..   RMSALAST - File Append Lock Blocking ASTs
 XDWM..   RMSALCNV - File Append Lock Conversions
 XDWM..   RMSALDEQ - File Append Lock Dequeues
 XDWM..   RMSALENQ - File Append Lock Enqueues
 XDWM..   RMSBUCKT - File Bucket Splits
 XDWM..E  RMSCLOSE - File Closes
 XDWM..E  RMSCONN  - File Connects
 XDWM..E  RMSDELET - File Deletes
 XDWM..E  RMSDISCN - File Disconnects
 XDWM..E  RMSEXTBL - File Extend Blocks
 XDWM..E  RMSEXTEN - File Extends
 XDWM..   RMSFBAST - File Lock Blocking ASTs
 XDWM..E  RMSFLUSH - File Flushes
 XDWM..   RMSGBAST - File Global Buffer Blocking ASTs
 XDWM..   RMSGBCNV - File Global Buffer Conversions
 XDWM..   RMSGBDEQ - File Global Buffer Dequeues
 XDWM..   RMSGBENQ - File Global Buffer Enqueues
 XDWM..   RMSGBRIO - File Global Buffer Read I/Os
 XDWM..   RMSGBWIO - File Global Buffer Write I/Os
 XDWM..   RMSGCATT - File Global Cache Attempts
 XDWM..   RMSGCHIT - File Global Cache Hits
 XDWM..E  RMSGETBT - File GET Bytes
 XDWM..   RMSGSCNV - File Global Section Conversions
 XDWM..   RMSGSDEQ - File Global Section Dequeues
 XDWM..   RMSGSENQ - File Global Section Enqueues
 XDWM..E  RMSKEYFD - Keyed Finds
 XDWM..E  RMSKEYGT - Keyed GETs
 XDWM..E  RMSKEYPT - Keyed PUTs
 XDWM..   RMSLBAST - File Local Buffer Blocking ASTs
 XDWM..   RMSLBCNV - File Local Buffer Conversions
 XDWM..   RMSLBDEQ - File Local Buffer Dequeues
 XDWM..   RMSLBENQ - File Local Buffer Enqueues
 XDWM..   RMSLBRIO - File Local Buffer Read I/Os
 XDWM..   RMSLBWIO - File Local Buffer Write I/Os
 XDWM..   RMSLCATT - File Local Cache Attempts
 XDWM..   RMSLCHIT - File Local Cache Hits
 XDWM..   RMSLKCNV - File Lock Conversions
 XDWM..   RMSLKDEQ - File Lock Dequeues
 XDWM..   RMSLKENQ - File Lock Enqueues
 XDWM..   RMSMULTB - File Multibucket Splits
 XDWM..E  RMSOPEN  - File Opens
 XDWM..E  RMSPUTBT - File PUT Bytes
 XDWM..E  RMSRDBT  - File Read Bytes
 XDWM..E  RMSREAD  - File Reads
 XDWM..E  RMSREWND - File Rewinds
 XDWM..E  RMSRFAFD - Record File Address Finds
 XDWM..E  RMSRFAGT - Record File Address GETs
 XDWM..   RMSRLCNV - Record Lock Conversions
 XDWM..   RMSRLDEQ - Record Lock Dequeues
 XDWM..   RMSRLENQ - Record Lock Enqueues
 XDWM..E  RMSSEQFD - Sequential Finds
 XDWM..E  RMSSEQGT - Sequential GETs
 XDWM..E  RMSSEQPT - Sequential PUTs
 XDWM..E  RMSTRUBL - File Truncate Blocks
 XDWM..E  RMSTRUNC - File Truncates
 XDWM..E  RMSUPDBT - File Update Bytes
 XDWM..E  RMSUPDTE - File Updates
 XDWM..E  RMSWRBT  - File Write Bytes
 XDWM..E  RMSWRITE - File Writes
```

5.3.5.3 - DEMRMS Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   - o HOUR should not be used in MONTHS and YEARS.
   - o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   - o WEEK should not be used in MONTHS or YEARS.
   - o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DEMRMSnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. This data element should only be referenced when using the DEMRMSnn file in the DETAIL timespan.

   RMSORG - Organization

5.3.5.4 - DEMRMS Retrieval Examples

This section presents typical DEMRMS retrieval examples.

1. Print the average number of bytes transferred per read I/O yesterday on node DEV1.

```
DATA;
SET &pDEMX..DEMRMS01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
RETAIN BYTES READS 0;
BYTES+RMSRDBT;
READS+RMSREAD;
IF EOF THEN DO;
  BYTESIO=BYTES/READS;
  PUT BYTESIO;
  END;
```

2. Print the total number of file opens yesterday on node DEV1.

```
DATA;
SET &pDEMX..DEMRMS01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
RETAIN OPENS 0;
OPENS+RMSOPEN;
IF EOF THEN DO;
  PUT OPENS;
  END;
```

5.3.6 - VMS System Profile File (DEMSPR)

The VMS System Profile file maintains consolidated information on many aspects of system operation. It is derived from the VMS MONITOR class records for system class, including type 1, 3, 4, 5, 6, 7, 8, 11, 14, 17, 19, 21, and 22 records.

This file contains information on:

- o Overall system operation, including CPU busy, read I/Os, direct I/Os, buffered I/Os, free page count, modified page count, and number of active processes.
- o Cluster-wide VMS lock activity, including enqueues, dequeues, and conversions.
- o DECnet activity, including arriving and departing packets, lost packets, and buffer failures.
- o Distributed Lock Management Facility, including enqueues, dequeues, and conversions.
- o VMS file system ancillary control processes (ACPs), including new files, page faults, and read and write I/Os.
- o VMS File System Cache, including cache attempts and hits for directories, file headers, file IDs, and extents.
- o VMS I/O subsystem, including direct I/Os, buffered I/Os, page reads, page writes, and page faults.
- o Lock management in VMS, including enqueues, dequeues, and deadlocks.
- o MSCP server activities, including reads, writes, requests, and I/Os of various sizes.
- o VMS Page Management System, including page faults, reads, read I/Os, writes, write I/Os, faults, and free pages.
- o VMS space allocation in the nonpaged dynamic pool, including packets free and in use (small, intermediate, and large), kilobytes free and in use, and largest and smallest blocks available.
- o Number of processes in each scheduler state: waiting, hibernating, suspended, inswapped, and outswapped.
- o Transaction processing statistics for DECdtm services, including transaction starts, commits, aborts, and ends.

The following sections describe the file's organization, and list the data elements maintained.

1 - DEMSPR File Organization
2 - DEMSPR Data Elements List
3 - DEMSPR Usage Considerations
4 - DEMSPR Retrieval Examples

5.3.6.1 - DEMSPR File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID ENDTS	YEAR	MONTH	DAY	HOUR
DAYS	SYSID	YEAR	MONTH	DAY	HOUR
WEEKS	SYSID	YEAR	WEEK	ZONE	HOUR
MONTHS	SYSID	YEAR	MONTH	ZONE	
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-12. DEMSPR Time-Span Granularity Chart

5.3.6.2 - DEMSPR Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)

Sequence/Summary Data Elements

XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
XDWM..E	ZONE	- Time Zone

Common Data Elements

XD....E	DAYNAME	- Name of Day of Week
X.....E	DEXMONVR	- VMS Monitor Version
XDWM..E	DURATION	- Recording Interval Time
XDWM..E	ENDTS	- End Time Stamp
XDWM..E	INTERVLS	- Number of Recording Intervals
X.....E	ORGSYSID	- Originating System Identification
XDWM..E	STARTTS	- Start Time Stamp

Retained Data Elements

X.....	SPRBIGHL	- Largest Block Unused
X.....	SPRCEF	- Common Event Flag Wait
X.....	SPRCOLPG	- Collided Page Wait
X.....E	SPRCOM	- Compute State Wait, Inswapped
X.....E	SPRCOMO	- Compute State Wait, Outswapped
X.....E	SPRCUR	- Current Processes
X.....E	SPRCURLK	- Current Locks
X.....E	SPRCURRS	- Current Resources
X.....	SPRFPGWT	- Free Page Wait
X.....E	SPRFRE	- Free Page List Size
X.....E	SPRFREEP	- IO Free Page Count
X.....	SPRFRPG	- Free Page Count
X.....E	SPRFRPGC	- PGE Free Page Count
X.....	SPRHIB	- Hibernate Wait, Inswapped
X.....	SPRHIBO	- Hibernate Wait, Outswapped
X.....	SPRHOLES	- Nonpaged Pool Unused KBytes
X.....E	SPRIPFRE	- Intermediate Request Packets Free
X.....E	SPRIPUSE	- Intermediate Request Packets in Use
X.....	SPRKBUSE	- Nonpaged Pool KBytes in Use
X.....	SPRLEF	- Local Event Flag Wait, Inswapped
X.....	SPRLEFO	- Local Event Flag Wait, Outswapped
X.....E	SPRLPFRE	- Large Request Packets Free
X.....E	SPRLPUSE	- Large Request Packets in Use
X.....E	SPRLRGPK	- Large Request Packets Left
X.....E	SPRMASK	- Record Construction Audit Mask
X.....	SPRMDPG	- Modified Page Count
X.....E	SPRMDPGC	- PGE Modified Page Count
X.....E	SPRMODP	- IO Modified Page Count
X.....	SPRMWAIT	- Miscellaneous Resource Wait
X.....	SPROSTAT	- Other States
X.....	SPRPFW	- Page Fault Wait
X.....E	SPRPRCNT	- Process Count
X.....	SPRSMBLK	- Block Less Than or Equal to 32
X.....	SPRSMHL	- Smallest Block Unused
X.....E	SPRSPFRE	- Small Request Packets Free
X.....E	SPRSPUSE	- Small Request Packets in Use
X.....	SPRSUSP	- Suspended Wait, Inswapped
X.....	SPRSUSPO	- Suspended Wait, Outswapped
X.....	SPRUNCON	- Nonpaged Pool Unused Contiguous Space

Accumulated Data Elements

XDWM..	SPRACCS	- File Access
XDWM..E	SPRALLOC	- Disk Allocations
XDWM..E	SPRARLPK	- Arriving Local Packets
XDWM..E	SPRARTPK	- Arriving Transit Packets
XDWM..	SPRBAI	- Blocking ASTs, Incoming
XDWM..	SPRBAL	- Blocking ASTs, Local
XDWM..	SPRBAO	- Blocking ASTs, Outgoing
XDWM..E	SPRBIOS	- Buffered I/Os
XDWM..	SPRBITAT	- Storage Bitmap Cache Attempts
XDWM..	SPRBITHT	- Storage Bitmap Cache Hits
XDWM..E	SPRBLAST	- Blocking ASTs
XDWM..	SPRBUFFL	- Receiver Buffer Failures
XDWM..E	SPRBUFIO	- Buffered I/Os
XDWM..E	SPRBWAIT	- MSCP Buffer Waits
XDWM..E	SPRCALLS	- FCP Calls
XDWM..E	SPRCENQ	- Converted Enqueues
XDWM..E	SPRCLSCT	- CLS CLUSTER Record Count (Type 19)
XDWM..E	SPRCLSDR	- CLS CLUSTER Record Duration (Type 19)
XDWM..E	SPRCLSTM	- CLS CPU Busy Time
XDWM..E	SPRCNTAB	- Transactions Aborted Count
XDWM..E	SPRCNTAD	- Add Remote Branch Ops Count
XDWM..E	SPRCNTBR	- Start Remote Branch Ops Count
XDWM..E	SPRCNTCO	- Transactions Committed Count
XDWM..	SPRCNTD1	- Transactions < 1 Second Count
XDWM..	SPRCNTD2	- Transactions 1-2 Seconds Count
XDWM..	SPRCNTD3	- Transactions 2-3 Seconds Count
XDWM..	SPRCNTD4	- Transactions 3-4 Seconds Count
XDWM..	SPRCNTD5	- Transactions 4-5 Seconds Count
XDWM..	SPRCNTD6	- Transactions > 5 Seconds Count
XDWM..E	SPRCNTEN	- Transactions Ended Count
XDWM..E	SPRCNTOP	- One-Phase Commits Initiated
XDWM..E	SPRCNTPR	- Transactions Prepared Count
XDWM..E	SPRCNTST	- Transactions Started Count
XDWM..E	SPRCPUNI	- Number of Instructions Executed
XDWM..E	SPRCPUTM	- System CPU Busy Time
XDWM..	SPRDATAT	- Directory Data Cache Attempts
XDWM..	SPRDATHT	- Directory Data Cache Hits
XDWM..E	SPRDEADF	- Deadlocks Found
XDWM..E	SPRDEADS	- Deadlock Searches
XDWM..E	SPRDEQ	- Dequeues
XDWM..E	SPRDIOS	- Direct I/Os
XDWM..E	SPRDIRIO	- Direct I/Os
XDWM..E	SPRDLKCT	- DLK DLOCK Record Count (Type 14)
XDWM..E	SPRDLKDR	- DLK DLOCK Record Duration (Type 14)
XDWM..	SPRDLM	- Deadlock Messages
XDWM..E	SPRDMZFL	- Demand Zero Page Faults
XDWM..E	SPRDNTCT	- DNT DECNET Record Count (Type 8)
XDWM..E	SPRDNTDR	- DNT DECNET Record Duration (Type 8)
XDWM..E	SPRDPLPK	- Departing Local Packets
XDWM..	SPRDQI	- Dequeue Incoming

```
XDWM..   SPRDQL   - Dequeue Local
XDWM..   SPRDQO   - Dequeue Outgoing
XDWM..   SPRECI   - Enqueue Conversions Incoming
XDWM..   SPRECL   - Enqueue Conversions Local
XDWM..   SPRECO   - Enqueue Conversions Outgoing
XDWM..E  SPRENQNQ - Enqueues Not Queued
XDWM..E  SPRENQWT - Enqueue Waits
XDWM..E  SPRERASE - Erase I/O Operations
XDWM..   SPREXTAT - Extent Cache Attempts
XDWM..   SPREXTHT - Extent Cache Hits
XDWM..   SPRFCBAT - Directory FCB Cache Attempts
XDWM..   SPRFCBHT - Directory FCB Cache Hits
XDWM..E  SPRFCPCT - FCP Record Count (Type 5)
XDWM..E  SPRFCPDR - FCP Record Duration (Type 5)
XDWM..   SPRFCPFL - FCP Page Faults
XDWM..E  SPRFCPRI - FCP Disk Read I/Os
XDWM..E  SPRFCPTM - FCP CPU Time
XDWM..E  SPRFCPWI - FCP Disk Write I/Os
XDWM..   SPRFHDAT - File Header Cache Attempts
XDWM..   SPRFHDHT - File Header Cache Hits
XDWM..   SPRFIDAT - File ID Cache Attempts
XDWM..   SPRFIDHT - File ID Cache Hits
XDWM..E  SPRFILOP - Files Opened
XDWM..E  SPRFNI   - Directory Functions Incoming
XDWM..E  SPRFNO   - Directory Functions Outgoing
XDWM..E  SPRFPGFL - Free List Page Faults
XDWM..E  SPRFRGMT - MSCP Fragments
XDWM..E  SPRFSCCT - FSC CACHE Record Count (Type 11)
XDWM..E  SPRFSCDR - FSC CACHE Record Duration (Type 11)
XDWM..E  SPRGBLFL - Global Valid Page Faults
XDWM..E  SPRINSWP - Inswaps
XDWM..E  SPRIOVCT - IOV IO Record Count (Type 4)
XDWM..E  SPRIOVDR - IOV IO Record Duration (Type 4)
XDWM..E  SPRIOVFL - IO Page Faults
XDWM..   SPRLCI   - Lock Conversions, Incoming
XDWM..   SPRLCL   - Lock Conversions, Local
XDWM..   SPRLCO   - Lock Conversions, Outgoing
XDWM..E  SPRLOGNM - Logical Name Translations
XDWM..E  SPRLOK   - Total All Locks
XDWM..E  SPRLOKCT - LOK LOCK Record Count (Type 7)
XDWM..E  SPRLOKDR - LOK LOCK Record Duration (Type 7)
XDWM..   SPRMAILW - Mailbox Writes
XDWM..E  SPRMPGFL - Modified List Page Faults
XDWM..E  SPRMSCCT - MSC MSCP Record Count (Type 21)
XDWM..E  SPRMSCDR - MSC MSCP Record Duration (Type 21)
XDWM..E  SPRMSCRD - MSCP Reads
XDWM..E  SPRMSCSP - MSCP Splits
XDWM..E  SPRMSCWR - MSCP Writes
XDWM..   SPRNEI   - New Enqueue Incoming
XDWM..   SPRNEL   - New Enqueue Local
XDWM..E  SPRNENQ  - New Enqueues
XDWM..   SPRNEO   - New Enqueue Outgoing
XDWM..E  SPRNEWFL - New Files
XDWM..   SPRNLI   - New Locks, Incoming
XDWM..   SPRNLL   - New Locks, Local
XDWM..   SPRNLO   - New Locks, Outgoing
XDWM..E  SPROPEN  - Files Opened
XDWM..E  SPRPGECT - PGE PAGE Record Count (Type 3)
XDWM..E  SPRPGEDR - PGE PAGE Record Duration (Type 3)
XDWM..E  SPRPGEFL - PGE Page Faults
XDWM..E  SPRPGERD - PGE Page Reads
XDWM..E  SPRPGERI - PGE Page Read I/Os
XDWM..E  SPRPGEWI - PGE Page Write I/Os
XDWM..E  SPRPGEWR - PGE Page Writes
XDWM..E  SPRPGFLT - Page Faults
XDWM..E  SPRPGRD  - Page Reads
XDWM..E  SPRPGWT  - Page Writes
XDWM..   SPRPKLOS - Transit Packets Lost
XDWM..E  SPRPOLCT - POL POOL Record Count (Type 6)
XDWM..E  SPRPOLDR - POL POOL Record Duration (Type 6)
XDWM..E  SPRPRDIO - Page Read I/Os
XDWM..E  SPRPWIO  - Page Write I/Os
XDWM..   SPRQUOAT - Quota Cache Attempts
XDWM..   SPRQUOHT - Quota Cache Hits
XDWM..E  SPRRDIOS - Page Read I/Os
XDWM..E  SPRREQST - MSCP Requests
XDWM..   SPRSIZE1 - 1 Block I/Os
XDWM..   SPRSIZE2 - 2-3 Block I/Os
XDWM..   SPRSIZE3 - 4-7 Block I/Os
XDWM..   SPRSIZE4 - 8-15 Block I/Os
XDWM..   SPRSIZE5 - 16-31 Block I/Os
XDWM..   SPRSIZE6 - 32-63 Block I/Os
XDWM..   SPRSIZE7 - 64 And Larger Block I/Os
XDWM..   SPRSPLIT - Split Transfers
XDWM..E  SPRSTACT - STA STATES Record Count (Type 1)
XDWM..E  SPRSTADR - STA STATES Record Duration (Type 1)
XDWM..E  SPRSYSFL - System Page Faults
XDWM..E  SPRSYTCT - SYT SYSTEM Record Count (Type 17)
XDWM..E  SPRSYTDR - SYT SYSTEM Record Duration (Type 17)
XDWM..E  SPRTRNCT - TRN TRANS Record Count (Type 22)
XDWM..E  SPRTRNDR - TRN TRANS Record Duration (Type 22)
XDWM..   SPRTURN  - Window Turns
XDWM..   SPRUNI   - Unlocks, Incoming
XDWM..   SPRUNL   - Unlocks, Local
XDWM..   SPRUNO   - Unlocks, Outgoing
XDWM..   SPRVLOK  - Volume Lock Waits
XDWM..E  SPRWIPFL - Write In Progress Page Faults
XDWM..   SPRWTURN - Window Turns
```

Minimum Data Elements

```
XDWM..   SPRMNFP  - Minimum Free Page Count
XDWM..E  SPRMNFRE - Minimum Free Page List Size
XDWM..   SPRMNHOL - Minimum Unused KBytes
```

```
| XDWM..E  SPRMNIFP - IO Minimum Free Page Count                  |
| XDWM..E  SPRMNIMP - IO Minimum Modified Page Count              |
| XDWM..E  SPRMNIPF - Minimum Intermediate Req Packets Free       |
| XDWM..   SPRMNLBK - Minimum Largest Block                       |
| XDWM..E  SPRMNLPF - Minimum Large Request Packets Free          |
| XDWM..E  SPRMNLPK - Minimum Large Request Packets Left          |
| XDWM..   SPRMNMP  - Minimum Modified Page Count                 |
| XDWM..E  SPRMNPFP - PGE Minimum Free Page Count                 |
| XDWM..E  SPRMNPMP - PGE Minimum Modified Page Count             |
| XDWM..   SPRMNSBK - Minimum Smallest Block                      |
| XDWM..E  SPRMNSPF - Minimum Small Request Packets Free          |
| XDWM..   SPRMNUCN - Minimum Unused Contiguous Space             |
|                                                                 |
| Maximum Data Elements                                           |
|                                                                 |
| XDWM..   SPRMSTAB - Max Transaction Abort Rate                  |
| XDWM..   SPRMSTAD - Max Transaction Remote Add Rate             |
| XDWM..   SPRMSTBR - Max Transaction Remote Start Rate           |
| XDWM..   SPRMSTCO - Max Transaction Total Commit Rate           |
| XDWM..   SPRMSTEN - Max Transaction End Rate                    |
| XDWM..   SPRMSTOP - Max Transaction One-Phase Init Rate         |
| XDWM..   SPRMSTPR - Max Transaction Prepare Rate                |
| XDWM..   SPRMSTST - Max Transaction Start Rate                  |
| XDWM..   SPRMXCEF - Maximum Common Event Flag Wait              |
| XDWM..E  SPRMXCLK - Maximum Current Locks                       |
| XDWM..E  SPRMXCP  - Maximum Current Processes                   |
| XDWM..   SPRMXCPW - Maximum Collided Page Wait                  |
| XDWM..E  SPRMXCRS - Maximum Current Resources                   |
| XDWM..E  SPRMXCSI - Maximum Compute State Wait, Inswapped       |
| XDWM..E  SPRMXCSO - Maximum Compute State Wait, Outswapped      |
| XDWM..   SPRMXFPW - Maximum Free Page Wait                      |
| XDWM..   SPRMXHWI - Maximum Hibernate Wait, Inswapped           |
| XDWM..   SPRMXHWO - Maximum Hibernate Wait, Outswapped          |
| XDWM..E  SPRMXIPU - Maximum Intermediate Req Packet in Use      |
| XDWM..   SPRMXKBU - Maximum Nonpaged Pool KBytes in Use         |
| XDWM..   SPRMXLEI - Maximum Local Event Flag Wait, Inswap       |
| XDWM..   SPRMXLEO - Maximum Local Event Flag Wait, Outswap      |
| XDWM..E  SPRMXLPU - Maximum Large Request Packets in Use        |
| XDWM..   SPRMXMRW - Maximum Miscellaneous Resource Wait         |
| XDWM..E  SPRMXOST - Maximum Other States                        |
| XDWM..   SPRMXPFW - Maximum Page Fault Wait                     |
| XDWM..E  SPRMXPRC - Maximum Process Count                       |
| XDWM..   SPRMXSMB - Maximum Block Less Than or Equal to 32      |
| XDWM..   SPRMXSPI - Maximum Suspended Wait, Inswapped           |
| XDWM..   SPRMXSPO - Maximum Suspended Wait, Outswapped          |
| XDWM..E  SPRMXSPU - Maximum Small Request Packets in Use        |
| XDWM..E  SPRTRNGE - DEXTS Range From Input Records              |
|                                                                 |
| Derived Data Elements                                           |
|                                                                 |
| XDWM..   SPRPCBIT - Storage Bitmap Cache Hit Percentage         |
| XDWM..   SPRPCDDC - Directory Data Cache Hit Percentage         |
| XDWM..   SPRPCEXT - Extent Cache Hit Percentage                 |
| XDWM..   SPRPCFCB - Directory FCB Cache Hit Percentage          |
| XDWM..   SPRPCFHC - File Header Cache Hit Percentage            |
| XDWM..   SPRPCFID - File ID Cache Hit Percentage                |
| XDWM..   SPRPCQOT - Quota Cache Hit Percentage                  |
| XDWM..   SPRPSACC - File Access Rate                            |
| XDWM..E  SPRPSALL - Disk Allocation Rate                        |
| XDWM..E  SPRPSARL - Arriving Local Packet Rate                  |
| XDWM..E  SPRPSAST - Blocking ASTs Rate                          |
| XDWM..E  SPRPSATP - Arriving Transit Packet Rate                |
| XDWM..   SPRPSBAI - Blocking ASTs Incoming Rate                 |
| XDWM..   SPRPSBAL - Blocking ASTs Local Rate                    |
| XDWM..   SPRPSBAO - Blocking ASTs Outgoing Rate                 |
| XDWM..E  SPRPSBIO - Buffered I/O Rate                           |
| XDWM..   SPRPSBIT - Storage Bitmap Cache Attempt Rate           |
| XDWM..E  SPRPSBUF - Buffered I/O Rate                           |
| XDWM..E  SPRPSCAL - FCP Call Rate                               |
| XDWM..E  SPRPSCEQ - Converted Enqueue Rate                      |
| XDWM..   SPRPSDDC - Directory Data Cache Attempt Rate           |
| XDWM..E  SPRPSDEQ - Dequeue Rate                                |
| XDWM..E  SPRPSDFD - Deadlocks Found Rate                        |
| XDWM..E  SPRPSDIO - Direct I/O Rate                             |
| XDWM..E  SPRPSDIR - Direct I/O Rate                             |
| XDWM..   SPRPSDLM - Deadlock Message Rate                       |
| XDWM..E  SPRPSDMZ - Demand Zero Page Fault Rate                 |
| XDWM..E  SPRPSDPL - Departing Local Packet Rate                 |
| XDWM..   SPRPSDQI - Dequeue Incoming Rate                       |
| XDWM..   SPRPSDQL - Dequeue Local Rate                          |
| XDWM..   SPRPSDQO - Dequeue Outgoing Rate                       |
| XDWM..E  SPRPSDSR - Deadlock Search Rate                        |
| XDWM..   SPRPSECI - Enqueue Conversion Incoming Rate            |
| XDWM..   SPRPSECL - Enqueue Conversion Local Rate               |
| XDWM..   SPRPSECO - Enqueue Conversion Outgoing Rate            |
| XDWM..E  SPRPSEIO - Erase I/O Rate                              |
| XDWM..E  SPRPSENQ - Enqueues Not Queued Rate                    |
| XDWM..E  SPRPSEWT - Enqueue Waits Rate                          |
| XDWM..   SPRPSEXT - Extent Cache Attempt Rate                   |
| XDWM..   SPRPSFCB - Directory FCB Cache Attempt Rate            |
| XDWM..   SPRPSFFL - FCP Page Fault Rate                         |
| XDWM..   SPRPSFHC - File Header Cache Attempt Rate              |
| XDWM..   SPRPSFID - File ID Cache Attempt Rate                  |
| XDWM..E  SPRPSFLT - Page Fault Rate                             |
| XDWM..E  SPRPSFNI - Directory Functions Incoming Rate           |
| XDWM..E  SPRPSFNO - Directory Functions Outgoing Rate           |
| XDWM..E  SPRPSFOP - Files Opened Rate                           |
| XDWM..E  SPRPSFPL - Free List Page Fault Rate                   |
| XDWM..E  SPRPSFRI - FCP Disk Read I/O Rate                      |
| XDWM..E  SPRPSFWI - FCP Disk Write I/O Rate                     |
| XDWM..E  SPRPSGBL - Global Valid Page Fault Rate                |
| XDWM..E  SPRPSIFL - IO Page Fault Rate                          |
| XDWM..E  SPRPSINS - Inswap Rate                                 |
| XDWM..   SPRPSLCI - Lock Conversion Incoming Rate               |
|                                                                 |
|                                                                 |
|                                                                 |
```

```
XDWM..   SPRPSLCL - Lock Conversion Local Rate
XDWM..   SPRPSLCO - Lock Conversion Outgoing Rate
XDWM..E  SPRPSLNT - Logical Name Translation Rate
XDWM..   SPRPSLOS - Transit Packet Loss Rate
XDWM..   SPRPSMBX - Mailbox Write Rate
XDWM..E  SPRPSMPL - Modified List Page Fault Rate
XDWM..   SPRPSNEI - New Enqueue Incoming Rate
XDWM..   SPRPSNEL - New Enqueue Local Rate
XDWM..   SPRPSNEO - New Enqueue Outgoing Rate
XDWM..E  SPRPSNEQ - New Enqueue Rate
XDWM..E  SPRPSNFL - New Files Rate
XDWM..   SPRPSNLI - New Locks Incoming Rate
XDWM..   SPRPSNLL - New Locks Local Rate
XDWM..   SPRPSNLO - New Locks Outgoing Rate
XDWM..E  SPRPSOPN - File Open Rate
XDWM..E  SPRPSPFL - PGE Page Fault Rate
XDWM..E  SPRPSPGR - Page Read Rate
XDWM..E  SPRPSPGW - Page Write Rate
XDWM..E  SPRPSPR  - Page Read I/O Rate
XDWM..E  SPRPSPRI - PGE Page Read I/O Rate
XDWM..E  SPRPSPW  - Page Write I/O Rate
XDWM..E  SPRPSPWI - PGE Page Write I/O Rate
XDWM..   SPRPSQOT - Quota Cache Attempt Rate
XDWM..   SPRPSRBF - Receiver Buffer Failure Rate
XDWM..E  SPRPSRED - PGE Page Read Rate
XDWM..E  SPRPSRIO - Read I/O Rate
XDWM..   SPRPSSPL - Split Transfer Rate
XDWM..E  SPRPSSYS - System Page Fault Rate
XDWM..E  SPRPSTAB - Transaction Abort Rate
XDWM..E  SPRPSTAD - Transaction Remote Add Rate
XDWM..E  SPRPSTBR - Transaction Remote Start Rate
XDWM..E  SPRPSTCO - Transaction Total Commit Rate
XDWM..E  SPRPSTEN - Transaction End Rate
XDWM..E  SPRPSTOP - Transaction One-Phase Init Rate
XDWM..E  SPRPSTPR - Transaction Prepare Rate
XDWM..   SPRPSTRN - Window Turn Rate
XDWM..E  SPRPSTST - Transaction Start Rate
XDWM..   SPRPSUNI - Unlock Incoming Rate
XDWM..   SPRPSUNL - Unlock Local Rate
XDWM..   SPRPSUNO - Unlock Outgoing Rate
XDWM..   SPRPSVLK - Volume Lock Wait Rate
XDWM..E  SPRPSWIP - Write In Progress Page Fault Rate
XDWM..E  SPRPSWRT - PGE Page Write Rate
```

5.3.6.3 - DEMSPR Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   o HOUR should not be used in MONTHS and YEARS.
   o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   o WEEK should not be used in MONTHS or YEARS.
   o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DEMSPRnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DEMSPRnn file in the DETAIL timespan.

```
SPRCOLPG  -  Collided Page Wait
SPRMWAIT  -  Miscellaneous Resource Wait
SPRCEF    -  Common Event Wait Flag
SPRPFW    -  Page Fault Wait
SPRLEF    -  Local Event Flag Wait, Inswapped
SPRLEFO   -  Local Event Flag Wait, Outswapped
SPRHIB    -  Hibernate Wait, Inswapped
SPRHIBO   -  Hibernate Wait, Outswapped
SPRSUSP   -  Suspended Wait, Inswapped
SPRSUSPO  -  Suspended Wait, Outswapped
SPRFPGWT  -  Free Page Wait
SPRCOM    -  Compute State Wait, Inswapped
SPRCOMO   -  Compute State Wait, Outswapped
SPRCUR    -  Current Processes
SPRFRPGC  -  Free Page Count
SPRMDPGC  -  Modified Page Count
SPRFREEP  -  Free Page Count
SPRMODP   -  Modified Page Count
SPRSPFRE  -  Small Request Packets Free
SPRSPUSE  -  Small Request Packets In Use
SPRIPFRE  -  Intermediate Request Packets Free
SPRIPUSE  -  Intermediate Request Packets In Use
SPRLPFRE  -  Large Request Packets Free
SPRLPUSE  -  Large Request Packets In Use
SPRHOLES  -  Unused KBytes
```

```
SPRKBUSE  -  KBytes In Use
SPRUNCON  -  Unused Contiguous Space
SPRBIGHL  -  Largest Block
SPRSMHL   -  Smallest Block
SPRSMBLK  -  Block Less Than Or Equal To 32
SPRCURLK  -  Current Locks
SPRCURRS  -  Current Resources
SPRLRGPK  -  Large Request Packets Left
SPROSTAT  -  Other States
SPRPRCNT  -  Process Count
SPRFRPG   -  Free Page Count
SPRMDPG   -  Modified Page Count
SPRFRE    -  Free List Size
```

4. File Content

The VMS System Profile file is constructed by joining the information from the following VMS MONITOR records.

- o VMS MONITOR STATES Record
- o VMS MONITOR PAGE Record
- o VMS MONITOR IO Record
- o VMS MONITOR FCP Record
- o VMS MONITOR POOL Record
- o VMS MONITOR LOCK Record
- o VMS MONITOR DECNET Record
- o VMS MONITOR FILE SYSTEM CACHE Record
- o VMS MONITOR DLOCK Record
- o VMS MONITOR SYSTEM Record
- o VMS MONITOR CLUSTER Record
- o VMS MONITOR MSCP SERVER Record
- o VMS MONITOR TRANSACTION Record

The data element SPRMASK (Record Construction Audit Mask) is a 13-byte element in the DEMSPR file, with each byte signifying the presence or absence of a particular record type used to construct each observation of the file. SPRMASK is meaningful only at the DETAIL level of summarization. See the data element description for more information on this field.

5.3.6.4 - DEMSPR Retrieval Examples

This section presents typical DEMSPR retrieval examples.

1. SYSTEM Record Elements

   o Print the minimum free page count during prime time (Zone 1) yesterday for each SYSID.

```
DATA;
SET &pDEMX..DEMSPR01; BY SYSID;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
IF ZONE=1;
RETAIN FPC;
IF FIRST.SYSID THEN DO;
  FPC=0;
  END;
FPC=MIN(SPRFRPG,FPC);
IF LAST.SYSID THEN DO;
  PUT FPC;
  END;
```

   o Print the maximum page fault rate during prime time (Zone 1) yesterday for each SYSID.

```
DATA;
SET &pDEMX..DEMSPR01; BY SYSID;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
IF ZONE=1;
RETAIN PGFLT;
IF FIRST.SYSID THEN DO;
  PGFLT=0;
  END;
PGFLT=MIN(SPRPSFLT,PGFLT);
IF LAST.SYSID THEN DO;
  PUT PGFLT;
  END;
```

2. STATES Record Elements

   o Print the ratio of all processes in the LEFO, SUSPO, and COMO process states yesterday.

```
DATA;
SET &pDEMX..DEMSPR01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
RETAIN LEFO SUSPO COMO CNTR 0;
LEFO+SPRLEFO;
SUSPO+SPRSUSPO;
```

```
COMO+SPRCOMO;
CNTR=CNTR+LEFO+SUSPO+COMO;
IF EOF THEN DO;
  IF CNTR GT 0 THEN DO;
    LEFO=LEFO/CNTR;
    SUSPO=SUSPO/CNTR;
    COMO=COMO/CNTR;
  END;
  PUT LEFO SUSPO COMO;
END;
```

o Print the percentage of processes in the MWAIT state.

```
DATA;
SET &pDEMX..DEMSPR01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
RETAIN MWAIT CNTR 0;
MWAIT+SPRMWAIT;
CNTR+SPRCUR;
IF EOF THEN DO;
  MWAIT=MWAIT/CNTR;
  PUT MWAIT;
  END;
```

3. PAGE Record Elements

o Plot the average page fault rate for node DEV1 yesterday during prime time (Zone 1).

```
DATA;
SET &pDEMX..DEMSPR01;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
IF ZONE=1;
PROC PLOT;
 PLOT SPRPSPFL*ENDTS;
```

o Print the ratio of the page faults to the sum of the page reads and page writes yesterday on node DEV1 between 9 am and noon.

```
DATA;
SET &pDEMX..DEMSPR01;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
IF HOUR GE 9 AND HOUR LT 12;
IF (SPRPGERD+SPRPGEWR) GT 0 THEN
  RATIO=SPRPGEFL/(SPRPGERD+SPRPGEWR);
PUT RATIO;
```

4. IO Record Elements

o Print the number of logical name translations yesterday between 3 pm and 5 pm on node DEV1.

```
DATA;
SET &pDEMD..DEMSPR01 END=EOF;
IF SYSID='DEV1';
IF DATEPART(ENDTS)=TODAY()-1;
IF HOUR GE 15 AND HOUR LT 17;
RETAIN LNT 0;
LNT+SPRLOGNM;
IF EOF THEN DO;
  PUT LNT;
  END;
```

o Plot the file open rate for node DEV1 for each hour yesterday.

```
DATA;
SET &pDEMD..DEMSPR01;
IF SYSID='DEV1';
IF DATEPART(ENDTS)=TODAY()-1;
PROC PLOT;
 PLOT SPRPSFOP*HOUR;
```

5. FCP Record Elements

o Print the average window turn rate for node SERV over the last week.

```
DATA;
SET &pDEMW..DEMSPR01 END=EOF;
IF SYSID='SERV';
RETAIN TRNDUR TRNSUM 0;
TRNDUR+DURATION;
TRNSUM+SPRTURN;
IF EOF THEN DO;
  TRNAVG=TRNSUM/TRNDUR;
  PUT TRNAVG;
  END;
```

o Print the average FCP call rate for nodes P1 and G1 during yesterday's prime time (zone 1).

```
DATA;
SET &pDEMX..DEMSPR01; BY SYSID;
IF SYSID='P1' OR SYSID='G1';
RETAIN XFCPCAL XFCPDUR;
IF FIRST.SYSID THEN DO;
  XFCPCAL=0;
```

```
        XFCPDUR=0;
        END;
      XFCPCAL+SPRCALLS;
      XFCPDUR+DURATION;
      IF LAST.SYSID THEN DO;
        AVGFCP=XFCPCAL/XFCPDUR;
        PUT AVGFCP;
        END;
```

6. POOL Record Elements

   o Print the minimum number of small, intermediate, and large request packets free throughout yesterday on node DEV1.

```
      DATA;
      SET &pDEMX..DEMSPR01 END=EOF;
      IF DATEPART(ENDTS)=TODAY()-1;
      IF SYSID='DEV1';
      RETAIN SRP IRP LRP 0;
      SRP=MIN(SPRSPFRE,SRP);
      IRP=MIN(SPRIPFRE,IRP);
      LRP=MIN(SPRLPFRE,LRP);
      IF EOF THEN DO;
        PUT SRP IRP LRP;
        END;
```

   o Print the average number of kilobytes of pool in use over the last week.

```
      DATA;
      SET &pDEMX..DEMSPR01 END=EOF;
      RETAIN KBPOOL CNTR 0;
      KBPOOL+SPRKBUSE;
      CNTR+SPRPOLCT;
      IF EOF THEN DO;
        KBPOOL=KBPOOL/CNTR;
        PUT KBPOOL;
        END;
```

7. LOCK Record Elements

   o Print the maximum enqueue wait rate during yesterday for node DEV1.

```
      DATA;
      SET &pDEMX..DEMSPR01 END=EOF;
      IF DATEPART(ENDTS)=TODAY()-1;
      IF SYSID='DEV1';
      RETAIN MAXENQW 0;
      MAXENQW=MAX(SPRPSEWT,MAXENQW);
      IF EOF THEN DO;
        PUT MAXENQW;
        END;
```

   o Print the number of new enqueues and converted enqueues by hour yesterday for node DEV1.

```
      DATA;
      SET &pDEMD..DEMSPR01 END=EOF;
      IF DATEPART(ENDTS)=TODAY()-1;
      IF SYSID='DEV1';
      PROC PRINT;
       VAR MONTH DAY YEAR HOUR SPRNENQ SPRCENQ;
```

8. DECNET Record Elements

   o Print the total number of packets lost yesterday for each hour of the day.

```
      DATA;
      SET &pDEMD..DEMSPR01;
      IF DATEPART(ENDTS)=TODAY()-1;
      PROC PRINT;
       VAR SYSID MONTH DAY YEAR HOUR SPRPKLOS;
```

   o Print the ratio of local packets arriving to local packets sent for each hour yesterday for node DEV1

```
      DATA;
      SET &pDEMD..DEMSPR01;
      IF DATEPART(ENDTS)=TODAY()-1;
      IF SYSID='DEV1';
      IF SPRDPLPK GT 0 THEN
        RATIO=SPRARLPK/SPRDPLPK;
      PROC PRINT;
       VAR SYSID MONTH DAY YEAR HOUR RATIO;
```

9. FSC Record Elements

   o Print the minimum extent cache hit percentage and quota cache hit percentage on node DEV1 during the last week.

```
      DATA;
      SET &pDEMW..DEMSPR01 END=EOF;
      IF SYSID='DEV1';
      RETAIN MINECHIT MINQCHIT 0;
      MINECHIT=MIN(SPRPCEXT,MINECHIT);
      MINQCHIT=MIN(SPRPCQOT,MINQCHIT);
      IF EOF THEN DO;
```

```
         PUT MINECHIT MINQCHIT;
         END;
```

o Plot the extent cache hit percentage for prime time (Zone 1) yesterday for each SYSID.

```
       DATA;
       SET &pDEMX..DEMSPR01;
       PROC PLOT;
        PLOT SPRPCEXT*ENDTS; BY SYSID;
```

10. DLOCK Record Elements

o Print the average deadlock message rate per hour over the last week.

```
       DATA;
       SET &pDEMW..DEMSPR01;
       PROC PRINT;
        VAR SYSID WEEK YEAR HOUR SPRPSDLM;
```

o Print the ratio of local new locks to incoming new locks per hour over the last week.

```
       DATA;
       SET &pDEMW..DEMSPR01;
       IF SPRNLI GT 0 THEN
         RATIO=SPRNLL/SPRNLI;
       PROC PRINT;
        VAR SYSID WEEK HOUR RATIO;
```

11. CLUSTER Record Elements

o Print the total number of enqueues and dequeues requested during yesterday's processing whenever CPU Busy was 90 percent or more.

```
       DATA;
       SET &pDEMX..DEMSPR01;
       IF DATEPART(ENDTS)=TODAY()-1;
       CPUBUSY=SPRCLSTM/DURATION;
       IF CPUBUSY GE 0.9;
       ENQUEUES=SPRNEL+SPRNEI+SPRNEO+SPRECL+SPRECI+SPRECO;
       DEQUEUES=SPRDQL+SPRDQI+SPRDQO;
       PROC PRINT;
        VAR SYSID ENDTS CPUBUSY ENQUEUES DEQUEUES;
```

o Plot the CPU time over prime time (Zone 1) yesterday for each SYSID.

```
       DATA;
       SET &pDEMX..DEMSPR01;
       IF DATEPART(ENDTS)=TODAY()-1;
       IF ZONE=1;
       PROC PLOT;
        PLOT SPRCLSTM*ENDTS; BY SYSID;
```

12. MSCP SERVER Record Elements

o Print the total number of reads and writes yesterday on node DEV1.

```
       DATA;
       SET &pDEMX..DEMSPR01 END=EOF;
       IF DATEPART(ENDTS)=TODAY()-1;
       IF SYSID='DEV1';
       RETAIN READS WRITES 0;
       READS+SPRMSCRD;
       WRITES+SPRMSCWR;
       IF EOF THEN DO;
         PUT READS WRITES;
         END;
```

o Print the times when buffer waits occurred yesterday on node DEV1.

```
       DATA;
       SET &pDEMX..DEMSPR01;END=EOF;
       IF DATEPART(ENDTS)=TODAY()-1;
       IF SYSID='DEV1';
       IF SPRBWAIT;
       PROC PRINT;
        VAR ENDTS SPRBWAIT;
```

13. TRANSACTION Record Elements

o Print the average transaction completion rates indexed by their duration categories, for each hour over the last week.

```
       DATA;
       SET &pDEMW..DEMSPR01;
        CRATE1=SPRCNTD1/DURATION;   /* < 1 second  */
        CRATE2=SPRCNTD2/DURATION;   /* 1-2 seconds */
        CRATE3=SPRCNTD3/DURATION;   /* 2-3 seconds */
        CRATE4=SPRCNTD4/DURATION;   /* 3-4 seconds */
        CRATE5=SPRCNTD5/DURATION;   /* 4-5 seconds */
        CRATE6=SPRCNTD6/DURATION;   /* > 5 seconds */
       PROC PRINT;
        VAR SYSID WEEK HOUR CRATE1-CRATE6;
```

5.4 - VAX/VMS System Usage Information Area Files

This section identifies each file in the VAX/VMS System Usage information area (DES), the file's organization, the data elements contained in the file, and the time-spans in which the data elements are supported.

Data elements are described in Appendix B, Data Dictionary.

This section describes the following files:

1 - VMS Disk Device File (DESDKD)
2 - VMS Disk Quota File (DESDKQ)
3 - VMS Disk Usage File (DESDKU)
4 - VMS Process Activity File (DESPRX)
5 - VMS System Status File (DESSYU)

5.4.1 - VMS Disk Device File (DESDKD)

The following sections describe the file's organization and list the data elements maintained.

1 - DESDKD File Organization
2 - DESDKD Data Elements List
3 - DESDKD Usage Considerations
4 - DESDKD Retrieval Examples

5.4.1.1 - DESDKD File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXDEVIC	DEXVOLNM	DKDDEVTY	DKDVSNAM
	YEAR	MONTH	DAY	ENDTS	
DAYS	SYSID	DEXDEVIC	DEXVOLNM	DKDDEVTY	DKDVSNAM
	YEAR	MONTH	DAY		
WEEKS	SYSID	DEXDEVIC	DEXVOLNM	DKDDEVTY	DKDVSNAM
	YEAR	WEEK			
MONTHS	SYSID	DEXDEVIC	DEXVOLNM	DKDDEVTY	DKDVSNAM
	YEAR	MONTH			
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-13. DESDKD Time-Span Granularity Chart

5.4.1.2 - DESDKD Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XDWM..E	DEXDEVIC	- Device Name
XDWM..E	DEXVOLNM	- Volume Name
XDWM..E	DKDDEVTY	- Device Type
XDWM..E	DKDVSNAM	- Volume Set Name
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year

```
 XDWM..E  YEAR     - Year of Century
```

Common Data Elements

```
 XD....E  DAYNAME  - Name of Day of Week
 X.....E  DEXNODE  - Nodename
 XDWM..E  DEXSUSTB - DEXSUS System Boot Time Stamp
 XDWM..E  DEXSUSTS - DEXSUS Collect Time Stamp
 X.....E  DEXVOLSR - Volume Serial Number
 XDWM..E  ENDTS    - End Time Stamp
 XDW...E  HOUR     - Hour of Day
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  ZONE     - Time Zone
```

Retained Data Elements

```
 XDWM..E  DKDBLKSF - Device Free Blocks
 XDWM..E  DKDBLKST - Device Total Blocks
 XDWM..E  DKDERRCT - Device Error Count
 XDWM..E  DKDVOFMT - Volume Format Type
 XDWM..E  DKDVOWNM - Volume Owner Name
```

5.4.1.3 - DESDKD Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   o HOUR should not be used in MONTHS and YEARS.
   o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   o WEEK should not be used in MONTHS or YEARS.
   o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DESDKDnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DESDKDnn file in the DETAIL timespan.

   DEXNODE - Nodename
   DEXVOLSR - Volume Serial Number

5.4.1.4 - DESDKD Retrieval Examples

This section presents a typical DESDKD retrieval example.

1. Print a listing of active disk devices on all VMS nodes showing volume name, device type, volume format, total blocks and blocks free, as of yesterday.

```
PROC PRINT DATA=&pDESX..DESDKD01;
VAR SYSID DEXDEVIC DEXVOLNM DKDDEVTY DKDVOFMT
    DKDBLKST DKDBLKSF ENDTS;
RUN;
```

5.4.2 - VMS Disk Quota File (DESDKQ)

The following sections describe the file's organization and list the data elements maintained.

1 - DESDKQ File Organization
2 - DESDKQ Data Elements List
3 - DESDKQ Usage Considerations
4 - DESDKQ Retrieval Examples

5.4.2.1 - DESDKQ File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXDEVIC	DKQUIC	YEAR	MONTH
	DAY	ENDTS			
DAYS	SYSID	DEXDEVIC	DKQUIC	YEAR	MONTH
	DAY				
WEEKS	SYSID	DEXDEVIC	DKQUIC	YEAR	WEEK
MONTHS	SYSID	DEXDEVIC	DKQUIC	YEAR	MONTH
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-14. DESDKQ Time-Span Granularity Chart

5.4.2.2 - DESDKQ Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)

Sequence/Summary Data Elements

XD....E	DAY	- Day of Month
XDWM..E	DEXDEVIC	- Device Name
XDWM..E	DKQUIC	- Disk Quota UIC
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century

Common Data Elements

```
 XD....E  DAYNAME  - Name of Day of Week
 X.....E  DEXNODE  - Nodename
 XDWM..E  DEXSUSTB - DEXSUS System Boot Time Stamp
 XDWM..E  DEXSUSTS - DEXSUS Collect Time Stamp
 XDWM..E  DURATION - Recording Interval Time
 XDWM..E  ENDTS    - End Time Stamp
 XDW...E  HOUR     - Hour of Day
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  STARTTS  - Start Time Stamp
 XDWM..E  ZONE     - Time Zone
```

Retained Data Elements

```
 X.....E  DKQOVLIM - Disk Quota Overdraft Limit (Blocks)
 X.....E  DKQPQUOT - Disk Quota Permanent Quota (Blocks)
```

Accumulated Data Elements

```
 XDWM..E  DKQMBHUS - Disk Quota Usage Megabyte (2**20) Hrs
 X.....E  DKQUSAGE - Disk Quota Usage (Blocks)
```

5.4.2.3 - DESDKQ Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   o HOUR should not be used in MONTHS and YEARS.
   o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   o WEEK should not be used in MONTHS or YEARS.
   o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DESDKQnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DESDKQnn file in the DETAIL timespan.

   DEXNODE - Nodename
   DKQOVLIM - Disk Quota Overdraft Limit (Blocks)
   DKQPQUOT - Disk Quota Permanent Quota (Blocks)
   DKQUSAGE - Disk Quota Usage (Blocks)

5.4.2.4 - DESDKQ Retrieval Examples

This section presents a typical DESDKQ retrieval example.

1. Print a list of disk quota usage, permanent quota, and megabyte hours of usage by VMS Diskquota-listed UIC for disk device DUA1 on node DEV1, as of yesterday.

```
DATA; SET &pDESX..DESDKQ01;
IF SYSID='DEV1';
IF DEXDEVIC='DUA2';
KEEP SYSID DKQUIC DKQUSAGE DKQPQUOT DKQMBHUS;
PROC PRINT;
RUN;
```

5.4.3 - VMS Disk Usage File (DESDKU)

The following sections describe the file's organization and list the data elements maintained.

1 - DESDKU File Organization
2 - DESDKU Data Elements List
3 - DESDKU Usage Considerations
4 - DESDKU Retrieval Examples

5.4.3.1 - DESDKU File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXDEVIC	DKUDIRTL	DKUDIR	DKUFILE
	YEAR	MONTH	DAY	ENDTS	DEXACT1
	DEXACT2	DEXACT3			
DAYS	SYSID	DEXDEVIC	DKUDIRTL	YEAR	MONTH
	DAY	DEXACT1	DEXACT2	DEXACT3	
WEEKS	N/A				
MONTHS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH				
YEARS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-15. DESDKU Time-Span Granularity Chart

5.4.3.2 - DESDKU Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XD.MY.E	DEXACT1	- DIVISION
XD.MY.E	DEXACT2	- UIC GROUP
XD.MY.E	DEXACT3	- USER NAME
XD....E	DEXDEVIC	- Device Name
X.....E	DKUDIR	- Directory Name
XD....E	DKUDIRTL	- Directory Top Level
X.....E	DKUFILE	- File Specification

```
 XD.M..E  MONTH    - Month of Year
 XD.MY.E  SYSID    - System Identifier
 XD....E  WEEK     - Week of Year
 XD.MY.E  YEAR     - Year of Century

Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 X.....E  DEXGRP   - User Identification Code Group
 X.....E  DEXMBR   - User Identification Code Member
 X.....   DEXNODE  - Nodename
 XD.MY.   DEXSUSTB - DEXSUS System Boot Time Stamp
 XD.MY.   DEXSUSTS - DEXSUS Collect Time Stamp
 X.....E  DEXVOLNM - Volume Name
 X.....E  DEXVOLSR - Volume Serial Number
 XD.MY.E  DURATION - Recording Interval Time
 XD.MY.E  ENDTS    - End Time Stamp
 XD....E  HOUR     - Hour of Day
 X.....E  ORGSYSID - Originating System Identification
 XD.MY.E  STARTTS  - Start Time Stamp
 XD.MY.E  ZONE     - Time Zone

Accumulated Data Elements

 X.....E  DKUBLKSA - Number of Blocks (512) Allocated
 X.....   DKUBLKST - Number of Blocks (512) Total
 X.....E  DKUBLKSU - Number of Blocks (512) Used
 XD.MY.E  DKUCNT   - File Count
 XD.MY.E  DKUCOST  - Disk Usage Charges
 XD.MY.E  DKUMBHA  - Megabyte (2**20) Hours Allocated
 XD.MY.E  DKUMBHU  - Megabyte (2**20) Hours Used
```

5.4.3.3 - DESDKU Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   o HOUR should not be used in MONTHS and YEARS.
   o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   o WEEK should not be used in MONTHS or YEARS.
   o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DESDKUnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DESDKUnn file in the DETAIL timespan.

   DKUBLKSA - Number of Blocks Allocated
   DKUBLKST - Number of Blocks Total
   DKUBLKSU - Number of Blocks Used

5.4.3.4 - DESDKU Retrieval Examples

This section presents a typical DESDKU retrieval example.

1. Print a list of disk usage, megabyte hours allocated and used, by VMS directory top level, for disk device DUA1 on node DEV1, as recorded yesterday.

```
%LET BY = SYSID DKUDIRTL YEAR MONTH DAY;
%LET BREAK=DAY;
DATA; SET %MFILE(TS=DA,DB=p,F=DKU01);
IF SYSID='DEV1';
IF DEXDEVIC='DUA1';
PROC SORT; BY &BY;
DATA; SET;
%DKUSUM;
KEEP &BY DKUMBHA DKUMBHU;
PROC PRINT;
RUN;
```

5.4.4 - VMS Process Activity File (DESPRX)

The following sections describe the file's organization and list the data elements maintained.

1 - DESPRX File Organization
2 - DESPRX Data Elements List
3 - DESPRX Usage Considerations
4 - DESPRX Retrieval Examples

5.4.4.1 - DESPRX File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	PRXTYPE	PRXPNAME	DEXUSER	DEXACCNT
	DEXGRP	DEXMBR	DEXTERM	YEAR	MONTH
	DAY	HOUR	ENDTS		
DAYS	SYSID	PRXTYPE	PRXPNAME	DEXUSER	DEXACCNT
	DEXGRP	DEXMBR	DEXTERM	YEAR	MONTH
	DAY	HOUR			
WEEKS	SYSID	PRXTYPE	PRXPNAME	DEXUSER	DEXACCNT
	DEXGRP	DEXMBR	DEXTERM	YEAR	WEEK
	ZONE	HOUR			
MONTHS	SYSID	PRXTYPE	PRXPNAME	DEXUSER	DEXACCNT
	DEXGRP	DEXMBR	DEXTERM	YEAR	MONTH
	ZONE				
YEARS	SYSID	PRXTYPE	PRXPNAME	DEXUSER	DEXACCNT
	DEXGRP	DEXMBR	DEXTERM	YEAR	ZONE
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-16. DESPRX Time-Span Granularity Chart

5.4.4.2 - DESPRX Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XDWMY.E	DEXACCNT	- Account Name
XDWMY.E	DEXGRP	- User Identification Code Group
XDWMY.E	DEXMBR	- User Identification Code Member
XDWMY.E	DEXTERM	- Terminal Name
XDWMY.E	DEXUSER	- Username
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year

```
 XDWMY.E  PRXPNAME - Process Name
 XDWMY.E  PRXTYPE  - Process Type
 XDWMY.E  SYSID    - System Identifier
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone

Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 XDWMY.   DEXBOOTS - Monitor System Boot Time Stamp
 XDWMY.   DEXDUR   - Requested Recording Interval Time
 X.....   DEXINTNO - Reserved Field Incremented by Monitor
 X.....   DEXMONVR - VMS Monitor Version
 X.....E  DEXPID   - Process ID
 XDWMY.   DEXTS    - Monitor Time Stamp
 XDWMY.E  DURATION - Recording Interval Time
 XDWMY.E  ENDTS    - End Time Stamp
 XDWMY.E  INTERVLS - Number of Recording Intervals
 X.....E  ORGSYSID - Originating System Identification
 XDWMY.E  STARTTS  - Start Time Stamp

Retained Data Elements

 X.....   PRXEVNTF - Event Flag Wait Mask
 X.....   PRXGLBPG - Global Page Count
 X.....   PRXGRP   - User Identification Code Group
 X.....   PRXIPID  - Internal Process ID
 X.....   PRXMBR   - User Identification Code Member
 X.....   PRXPRCPG - Process Page Count
 X.....   PRXPRTY  - Priority
 X.....   PRXSTATE - State
 X.....   PRXSTATF - Status Flags
 X.....   PRXWKSET - Process Working Set

Accumulated Data Elements

 XDWMY.E  PRXBUFIO - Buffered I/Os
 XDWMY.E  PRXCPUNI - Number of Instructions Executed
 XDWMY.E  PRXCPUTM - CPU Time
 XDWMY.E  PRXDIRIO - Direct I/Os
 XDWMY.E  PRXPGFLT - Page Faults

Maximum Data Elements

 XDWMY.E  PRXMXGPG - Maximum Global Page Count
 XDWMY.E  PRXMXPPG - Maximum Process Page Count
```

5.4.4.3 - DESPRX Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   - o HOUR should not be used in MONTHS and YEARS.
   - o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   - o WEEK should not be used in MONTHS or YEARS.
   - o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DESPRXnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DESPRXnn file in the DETAIL timespan.

```
 DEXPID    -  Process ID
 DEXINTNO  -  Reserved field incremented by Monitor
 DEXMONVR  -  VMS Monitor Version
 PRXEVNTF  -  Event Flag Wait Mask
 PRXGLBPG  -  Global Page Count
 PRXGRP    -  User Identification Code Group
 PRXIPID   -  Internal Process ID
 PRXMBR    -  User Identification Code Member
 PRXPNAME  -  Process Name
 PRXPRCPG  -  Process Page Count
 PRXPRTY   -  Priority
 PRXSTATE  -  State
 PRXSTATF  -  Status Flags
 PRXWKSET  -  Process Working Set
```

5.4.4.4 - DESPRX Retrieval Examples

This section presents typical DESPRX retrieval examples.

1. Print the total CPU time, direct I/Os, buffered I/Os, and page faults of all processes running during zone 1 hours 10 and 11, on system DEV1, by process name, user name and hour of day, for the past two weeks.

```
%LET BY = SYSID PRXPNAME DEXUSER YEAR WEEK ZONE HOUR;
%LET BREAK = HOUR;
DATA; SET %MFILE(TS=WE,DB=p,F=PRX02-01);
IF SYSID='DEV1';
IF ZONE EQ '1';
IF HOUR EQ 10 OR HOUR EQ 11;
PROC SORT; BY &BY;
DATA; SET;
%PRXSUM;
KEEP &BY PRXCPUTM PRXDIRIO PRXBUFIO PRXPGFLT;
PROC PRINT;
RUN;
```

5.4.5 - VMS System Status File (DESSYU)

The following sections describe the file's organization and list the data elements maintained.

1 - DESSYU File Organization
2 - DESSYU Data Elements List
3 - DESSYU Usage Considerations
4 - DESSYU Retrieval Examples

5.4.5.1 - DESSYU File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	YEAR	MONTH	DAY	ENDTS
DAYS	SYSID	YEAR	MONTH	DAY	
WEEKS	SYSID	YEAR	WEEK		
MONTHS	SYSID	YEAR	MONTH		
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-17. DESSYU Time-Span Granularity Chart

5.4.5.2 - DESSYU Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
Common Data Elements		

```
 XD....E  DAYNAME  - Name of Day of Week
 X.....E  DEXNODE  - Nodename
 XDWM..E  DEXSUSTB - DEXSUS System Boot Time Stamp
 XDWM..E  DEXSUSTS - DEXSUS Collect Time Stamp
 XDWM..E  ENDTS    - End Time Stamp
 XDW...E  HOUR     - Hour of Day
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  ZONE     - Time Zone
```

Retained Data Elements

```
 XDWM..E  SYUBALSM - Process Pages for Memory Allocation
 XDWM..E  SYUCPUS  - Number of CPUs
 XDWM..E  SYUHWNAM - Hardware Name
 XDWM..E  SYUIRPC  - IRP Lookaside List, Current Packets
 XDWM..E  SYUIRPI  - IRP Lookaside List, Initial Packets
 XDWM..E  SYUIRPM  - IRP Lookaside List, Maximum Packets
 XDWM..E  SYULRPC  - LRP Lookaside List, Current Packets
 XDWM..E  SYULRPI  - LRP Lookaside List, Initial Packets
 XDWM..E  SYULRPM  - LRP Lookaside List, Maximum Packets
 XDWM..E  SYUMEMAN - Main Memory Mb (physical)
 XDWM..E  SYUMENDC - Nonpaged Dynamic Memory, Current Size
 XDWM..E  SYUMENDI - Nonpaged Dynamic Memory, Initial Size
 XDWM..E  SYUMEPDC - Paged Dynamic Memory, Current Size
 XDWM..E  SYUMPWHI - MPW_HILIMIT System Parameter
 XDWM..E  SYUMXPRC - MAXPROCESSCNT System Parameter
 XDWM..E  SYUSRPC  - SRP Lookaside List, Current Packets
 XDWM..E  SYUSRPI  - SRP Lookaside List, Initial Packets
 XDWM..E  SYUSRPM  - SRP Lookaside List, Maximum Packets
 XDWM..E  SYUUPTIM - System Uptime
 XDWM..E  SYUVCPUS - Number of Vector-Present Processors
 XDWM..E  SYUVMSVR - VMS Version
```

Maximum Data Elements

```
 XDWM..E  SYUMXMEN - Nonpaged Dynamic Memory, Maximum Size
```

5.4.5.3 - DESSYU Usage Considerations

1. Care must be exercised in using the special date and time data elements contained in each CA MICS file. As the file's granularity increases in higher timespans, certain fields lose significance and should not be referenced:

   o HOUR should not be used in MONTHS and YEARS.
   o DAY and DAYNAME should not be used in WEEKS, MONTHS, or YEARS.
   o WEEK should not be used in MONTHS or YEARS.
   o MONTH should not be used in YEARS.

2. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan from when they are used in the DAYS, WEEKS, MONTHS, and YEARS timespans. The ENDTS and STARTTS, when appearing in the higher timespans, indicate the span of time over which the data has been summarized. STARTTS indicates the beginning of the timespan and ENDTS indicates the end of the timespan.

3. The following data elements only have meaning when using the DESSYUnn file in the DETAIL timespan, in that they lose significance once summarization has been performed. These data elements should only be referenced when using the DESSYUnn file in the DETAIL timespan.

```
 DEXNODE   -  Nodename
```

5.4.5.4 - DESSYU Retrieval Examples

This section presents typical DESSYU retrieval examples.

1. Print a list of Initial, Current, and Maximum Nonpaged Dynamic Memory readings from DEXSUS for the last seven days for node DEV1.

```
DATA;
SET %MFILE(TS=DA,DB=p,F=SYU07-01);
IF SYSID='DEV1';
PROC PRINT;
 VAR SYUMENDI SYUMENDC SYUMXMEN;
RUN;
```

2. Print a report of maximum VMS memory utilization by hour of day for the last seven days for node DEV1. This example merges the Minimum Free Page Count element (SPRMNIFP) from the System Profile (DEMSPR) file with the Main Memory element (SYUMEMAN) from the System Status (DESSYU) file to derive a memory utilization value.

```
%LET BY=SYSID YEAR MONTH DAY;
DATA SPR1;
 SET %MFILE(TS=DA,DB=p,F=SPR07-01);
 IF SYSID='DEV1';
 KEEP &BY HOUR SPRMNIFP;
PROC SORT; BY &BY HOUR;
DATA SYU1;
 SET %MFILE(TS=DA,DB=p,F=SYU07-01);
 IF SYSID='DEV1';
PROC SORT; BY &BY;
DATA MEM1; MERGE SPR1 SYU1;
 BY &BY;
 IF SYUMEMAN GT 0 THEN
  PCTMEMUS=(1-((SPRMNIFP*512)/(SYUMEMAN*(2**20))));
PROC PRINT DATA=MEM1;
 BY &BY; ID &BY;
 VAR HOUR PCTMEMUS SPRMNIFP SYUMEMAN SYUHWNAM;
 FORMAT PCTMEMUS PERCENT8.1 YEAR MONTH DAY HOUR 2.;
RUN;
```

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 6

DATA SOURCES

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 6

DATA SOURCES

Chapter 6. DATA SOURCES

The CA MICS VAX/VMS Analyzer processes data from VMS ACCOUNTING, VMS MONITOR, and the CA MICS VMS System Usage collection routine (DEXSUS).

This chapter contains the following sections:

1 - Data Collection and Transport to CA MICS
2 - VMS ACCOUNTING Record Descriptions
3 - VMS MONITOR Record Descriptions
4 - CA MICS VMS System Usage (DEXSUS) Record Descriptions

6.1 - Data Collection and Transport to CA MICS

The CA MICS VAX/VMS Analyzer processes data from three sources: VMS ACCOUNTING data, VMS MONITOR data, and CA MICS VMS System Usage data. The data from each of these three data sources is independent. VMS ACCOUNTING data is similar to MVS SMF data in that it is job or process oriented. VMS MONITOR data is similar to MVS RMF data in that it is interval oriented. CA MICS VMS System Usage data is a collection of measurements taken on a daily basis, according to the schedule you establish, typically once per day. Any or all of these three data sources can be used as input to the CA MICS VAX/VMS Analyzer. The most complete view of your system is gained by using all three data sources.

OBTAINING INPUT DATA FOR THE VAX/VMS ANALYZER

The CA MICS VAX/VMS Analyzer is designed to process VMS ACCOUNTING data in its original state as output to the Accounting log file and process VMS MONITOR data as output to a Monitor disk recording file in binary format. The Analyzer also processes CA MICS VMS System Usage (DEXSUS) data as written on the VMS system. The CA MICS VAX/VMS Analyzer performs the data conversions necessary for CA MICS processing so you should not attempt to translate the data before providing it as input to CA MICS. This fact should be taken into consideration when establishing a method for transporting the VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS data to your CA MICS system under MVS.

For most VAX processors, other than MicroVAX, Accounting is started automatically when the system is started. Accounting collects system usage statistics to a log file, normally called SYS$MANAGER:ACCOUNTNG.DAT. You can control the logging performed by Accounting using the SET ACCOUNTING command in DCL, if you have the privilege to do so. For instance, using this command, you can close the current log and open a new version.

To obtain VMS MONITOR data for CA MICS, you must initiate a background Monitor session that records to disk in binary format for the time period you wish to measure. You will need to set up a regular processing schedule on your VMS system to record this data on a daily basis.

To obtain DEXSUS data for CA MICS, you must invoke the DEXSUS.COM routine on a periodic basis, typically once per day. A sample routine called DEXSUSJ1.COM is supplied with the CA MICS VAX/VMS Analyzer to help you set up and schedule a batch job that records the DEXSUS data for CA MICS. You will need to set up a regular processing schedule on your VMS

system to record this data on a daily basis.

For all of these VMS data sources you will need to establish a procedure for transporting the data to your MVS system where the CA MICS VAX/VMS Analyzer operates. This can be done through a relatively manual process such as moving tapes from system to system each day. If another form of communication link is available between your VAX and your IBM systems, however, you may choose to transmit the VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS data over the link.

Whatever method you employ for getting the VMS data to CA MICS, be sure to obtain and transport the recording files in their original binary format and with a record format of variable (RECFM=VB) because this is the format expected by the CA MICS VAX/VMS Analyzer input routines.

The following sections discuss:

1 - Data Collector Operations
2 - Data Volume Considerations
3 - Transporting Data from VMS to MVS
4 - Errors Identified in Measurement Data

6.1.1 - Data Collector Operations

VMS ACCOUNTING and VMS MONITOR are standard data sources on VMS Systems. The sections that follow illustrate the typical operation of the ACCOUNTING and MONITOR utilities for use with CA MICS. The operation of these utilities is described fully in the related documentation offered by Compaq Computer Corporation.

The CA MICS VMS System Usage collection routine (DEXSUS) coordinates and combines the operation of several other VMS Utilities supported by Hewlett Packard Corporation, including:

VMS ANALYZE/DISK_STRUCTURE Utility

VMS SYSMAN Utility, specifically the DISKQUOTA SHOW command

VMS SHOW command

VMS DCL lexical functions

The CA MICS VMS System Usage collection routine is a Digital Command Language (DCL) routine called DEXSUS.COM. The sections that follow illustrate typical operation and explain the options of the DEXSUS.COM routine. Chapter 8 of this guide explains how to install the DEXSUS.COM routine.

The following sections discuss:

1 - VMS ACCOUNTING Data Collection
2 - VMS MONITOR Data Collection
3 - CA MICS VMS System Usage Data Collection (DEXSUS)

6.1.1.1 - VMS ACCOUNTING Data Collection

To start VMS ACCOUNTING, enter:

```
SET ACCOUNTING /NEW_FILE /ENABLE=(keywords)
```

where

NEW_FILE

specifies that the current VMS ACCOUNTING file should be closed and a new version opened.

'keywords'

are PROCESS and IMAGE. PROCESS is used if accounting at the process level is desired. IMAGE is used if accounting at the image level is desired. PROCESS,IMAGE is specified if both levels of accounting are desired.

PROCESS accounting writes a PROCESS TERMINATION record whenever a process terminates. IMAGE accounting writes an IMAGE TERMINATION record whenever an image executing within a process terminates. For example, a process is created for a user when the user logs on to the system. The user may then execute editors, compilers, linkers, system utilities, and previously developed application programs. In doing so, only one process has been created.

With PROCESS accounting, when the user logs off, a single PROCESS TERMINATION record will be created. With IMAGE accounting enabled, an IMAGE TERMINATION record will be created at the termination of each edit, compile, link, utility, or other program execution. It is clear that IMAGE accounting will create substantially more records than will PROCESS accounting. IMAGE accounting is useful when tracking the resource use of particular program executions is desired.

Each time the SET ACCOUNTING command is executed, any current ACCOUNTNG.DAT file is closed and a new one is started.

A full description of how to start the ACCOUNTING recording can be found in the VMS Systems Management Manual.

Sample Procedure:

At midnight, a command procedure is started on each cluster node that does a SET ACCOUNTING/NEW_FILE that closes the current file and opens a new version of ACCOUNTNG.DAT. The procedure then extracts the data from this file using the VMS Accounting Utility and issues a file transfer command to pass the extracted version of the file to MVS for processing in the CA MICS DAILY run.

An alternative is to use the VMS Accounting Utility each day to extract the previous day's data from ACCOUNTNG.DAT, if this file is kept as a cumulative file for a longer time period, such as an entire week or month.

For example, to extract yesterday's accounting data from ACCOUNTNG.DAT, issue the command:

```
ACCOUNTING /BINARY/LOG/NOREJECT/NOSORT -
 /OUTPUT=ACyymmdd.dat /SINCE=YESTERDAY /BEFORE=TODAY
```

See the VMS Accounting Utility Manual for a full description of how to use VMS Accounting to extract data.

6.1.1.2 - VMS MONITOR Data Collection

To start VMS MONITOR, enter:

```
MONITOR / RECORD=filename / NODISPLAY / INTERVAL=interval
        / ENDING=23:59   ALL_CLASSES
```

where

filename

is the name of the data set on which the VMS MONITOR data will be recorded.

NODISPLAY

indicates that a data file will be created rather than generating terminal displays.

interval

is the number of seconds between VMS MONITOR recordings.

It is difficult to specify a recording interval that is correct for all situations. In a fairly well-tuned system, an interval of 20 seconds is reasonable for ongoing performance monitoring. When analyzing system performance problems, the recording interval must be made much shorter. When using the PROCESS file for system diagnosis, a recording interval of 2 to 5 seconds is suggested. When using the MODES file, an interval of 10 seconds is appropriate. When analyzing file system problems with the RMS file, an interval of 1 to 3 seconds is required.

We recommend that you also consider the CA MICS database space requirements in choosing a reasonable interval value for data input to CA MICS. While an interval of 20 seconds may be useful for specific performance studies using VMS MONITOR in real-time, a more reasonable value for historical data summarization and retention purposes should be considered. Too small an interval generates so much data that its use is prohibitive. We suggest you start with a 10 or 15 minute interval (600 to 900 seconds) when setting up data collection for CA MICS use. You can adjust this setting after you have determined the data transfer and storage requirements from a low volume data flow.

ENDING=23:59

indicates that the recording should stop and the data file should be closed at midnight. The system operator should rename the file and restart VMS MONITOR.

ALL_CLASSES

indicates that all data record types should be created.

This command starts recording VMS MONITOR data in specified intervals until 23:59 and then stops recording. The binary file name in the RECORD subcommand is the file that will be passed over to MVS for the CA MICS DAILY run for VMS MONITOR data. When this file is passed to MVS, it must be in its original ASCII format (use binary transfer) and must be created on the MVS system with DCB=(RECFM=VB,LRECL=32752,BLKSIZE=32756).

Sample Procedure:

At midnight, submit a command procedure to each cluster node that cleans out a holding area, stops the VMS MONITOR, moves the VMS MONITOR files to the clean holding area and restarts the VMS MONITOR, and passes the data in the holding area to MVS for processing in the CA MICS daily run.

6.1.1.3 - CA MICS VMS System Usage Data Collection (DEXSUS)

The CA MICS VMS System Usage collection routine is a DCL routine called DEXSUS.COM. This section explains the execution options and typical operation of the DEXSUS.COM routine under VMS. Chapter 8 of this guide explains how to install the DEXSUS.COM routine on your VMS system.

The DEXSUS.COM routine collects three types of information about disk devices from the following VMS utilities:

- SHOW DEVICE/FULL
- DISKQUOTA
- ANALYZE/DISK_USAGE

You can control the operation of each of these utilities with the options described below. The options allow you to select which disk devices are measured by each of the three utilities. You can select or exclude certain devices. You can also select which utilities to run. Standard operation of the DEXSUS routine is to collect all three types of disk information on all active, mounted disk devices on the system, as determined by the command SHOW DEV D /MOUNTED. The DEXSUS routine writes all the data it collects to a single file named DEXSUS.DAT.

To start DEXSUS.COM, enter:

```
@DEXSUS
```

Options (VMS global symbol assignments)

```
DEXSUSDF - default directory for DEXSUS.COM operation

DEXSUSO1 - default options for SHOW DEV in DEXSUS.COM

DEXSUSL1 - disk select list for: SHOW/DEV/FULL
DEXSUSL2 - disk select list for: DISKQUOTA
DEXSUSL3 - disk select list for: ANALYZE/DISK

DEXSUSX1 - disk exclude list for: SHOW/DEV/FULL
DEXSUSX2 - disk exclude list for: DISKQUOTA
DEXSUSX3 - disk exclude list for: ANALYZE/DISK

DEXSUSV1 - volume set switch for: SHOW/DEV/FULL
DEXSUSV2 - volume set switch for: DISKQUOTA
DEXSUSV3 - volume set switch for: ANALYZE/DISK

DEXSUSS1 - shadow master switch for: SHOW/DEV/FULL
DEXSUSS2 - shadow master switch for: DISKQUOTA
DEXSUSS3 - shadow master switch for: ANALYZE/DISK

DEXSUSL  - disk select list: L1, L2, L3 default
DEXSUSX  - disk exclude list: X1, X2, X3 default
DEXSUSV  - volume set switch: V1, V2, V3 default
DEXSUSS  - shadow master switch: S1, S2, S3 default
```

The DEXSUSDF option sets the default directory for DEXSUS.COM operation. Temporary files and the completed DEXSUS.DAT output file are written to this directory.

The DEXSUSO1 option sets the criteria for the SHOW DEVICE command that determines the full list of active, mounted disk devices on the system. The default value is "D /MOUNTED", causing the SHOW DEV D /MOUNTED command to determine which disk devices to measure.

The DEXSUSLn and DEXSUSXn options allow you to control the collection of data from each of the three disk measurement utilities. For each type of disk data collection source, you can list a specific set of disk devices to select or exclude from the list of all active disks. The DEXSUS routine recognizes the presence of these lists, and limits collection activities accordingly.

The DEXSUSVn and DEXSUSSn options control the collection of data from Volume Sets and Shadow Sets by each of the three disk measurement utilities. By default, only Volume 1 of Volume Sets and only Shadow Masters in Shadow Sets are measured. This is indicated by a setting of "Y" (or "YES") for the DEXSUSVn and DEXSUSSn options. This means that other volumes in these sets are normally excluded from measurement. You can include all volumes in Volume Sets or Shadow Sets for each of the three disk measurement utilities by setting the appropriate DEXSUSVn and DEXSUSSn exclusion options to "N" (or "NO").

If you assign a value to DEXSUSx, then DEXSUSx1, DEXSUSx2, and DEXSUSx3 will all default to the value of DEXSUSx. However, if you also assign a value to DEXSUSx1, DEXSUSx2, or DEXSUSx3, then this assigned value overrides the default assigned to DEXSUSx for that specific list. 'x' can be one of the following characters: L, X, V, or S.

To completely deactivate the collection of data by any of the three utilities, use a select list of "X". For example, to completely skip DISKQUOTA processing use:

```
DEXSUSL2=="X"
```

REMINDER: Use VMS global assignments (==) to set the values

of any of the option symbols.

EXAMPLES

The following examples illustrate the use and syntax of the options for DEXSUS.COM execution.

```
Example:  DEXSUSDF=="<MICS.DEXSUS>"
          DEXSUSL2=="DUA1 DUA2"
          DEXSUSX3=="DUS1"
          @DEXSUS
```

In this example, DEXSUS will use the directory <MICS.DEXSUS> as its default directory, by issuing SET DEF 'dexsusdf'. The assignment of DEXSUSL2 will limit data collection for DISKQUOTA to the devices DUA1 and DUA2. The assignment of DEXSUSX3 will exclude device DUS1 from data collection by ANALYZE/DISK. All available mounted disks will be included in the SHOW DEVICE disk measurement function of DEXSUS because neither DEXSUSL1 nor DEXSUSX1 (nor DEXSUSL nor DEXSUSX) has been assigned to limit this function. However, only Volume 1 of Volume Sets and only Shadow Masters will be considered for SHOW DEVICE and ANALYZE/DISK because of the defaults taken for DEXSUSVn and DEXSUSSn.

```
Example:  DEXSUSDF=="<MICS.DEXSUS>"
          DEXSUSL2=="X"
          DEXSUSL3=="X"
          DEXSUSV1=="N"
          DEXSUSS1=="N"
          @DEXSUS
```

In this example, DEXSUS will use the directory <MICS.DEXSUS> as its default directory, and will scan all available mounted disks. The only disk measurements taken will be those from the SHOW DEVICE function of DEXSUS because DEXSUSL2 and DEXSUSL3 have been set to "X". Neither DISKQUOTA nor ANALYZE/DISK data will be collected. All disks, including all members of Volume Sets and all Shadow Members, will be included because the DEXSUSV1 and DEXSUSS1 filters have been disabled with a setting of "N".

SAMPLE JOB

A sample routine called DEXSUSJ1.COM is supplied with the CA MICS VAX/VMS Analyzer to help you set up and schedule a batch job that records the DEXSUS data for CA MICS. A copy of this routine is stored on your CA MICS system in sharedprefix.MICS.SOURCE(DEXSUSJ1).

EXAMPLE: @DEXSUSJ1 TOMORROW 999 X

Will run DEXSUS each day for 999 days. The X option causes the routine to bypass immediate interactive execution.

EXAMPLE: @DEXSUSJ1

Will run DEXSUS immediately, one time, and then shut down.

6.1.2 - Data Volume Considerations

ACCOUNTING DATA VOLUMES

VMS ACCOUNTING is a fairly low-volume data source. Records are produced only when processes and images terminate, when a LOGIN attempt fails, and when print jobs are processed.

Before enabling image-level accounting, you should consider the expense required in disk space and processing time. Image termination data can be useful for performance analysis but should be collected only in sufficient quantity to serve a specific purpose or study. Also, you may choose to collect image data for only certain image names through image install options. It is inadvisable to attempt full image-level accounting or even selected image accounting on a widely used image because the VMS overhead will be high.

If you choose to enable image-level accounting, review the data element definitions for the DEAIMG file in sharedprefix.MICS.GENLIB(DEXGENIN). The CA MICS data element IMGNAME is defined with a default length of 60 characters. You may want to change this length based on your recording requirements and the amount of database space needed to store this field. Refer to the CA MICS System Modification Guide for more information on database tailoring and GENIN members.

MONITOR DATA VOLUMES

VMS MONITOR can be a high-volume data source depending on the recording interval and the specific records that have been enabled. While an interval of 20 seconds may be useful for specific performance studies, a more reasonable value for historical data summarization and retention purposes should be considered. Too small an interval will actually generate so much data as to make its use prohibitive. We suggest you start with a 10 or 15 minute interval (600 to 900 seconds) when setting up data collection for CA MICS use. Once you have been able to determine the affect of this setting on data volume and data use, you can adjust the interval on the data collector.

The following records are written once per interval:

- CLUSTER
- DECNET
- DLOCK
- FCP
- FILE SYSTEM CACHE
- IO
- LOCK
- MSCP SERVER
- PAGE
- POOL
- STATES
- SYSTEM
- TRANSACTION

The following records contain multiple segments and generate multiple CA MICS records:

- DISK - One per each selected disk device per interval
- MODES - One per CPU per interval
- PROCESSES - One per process per interval
- RMS - One per file per interval
- SCS - One per node per interval
- VECTOR - One per CPU per interval

DEXSUS DATA VOLUMES

DEXSUS can be a high-volume data source depending on the frequency of operation and the specific system usage measurements that have been enabled.

The most voluminous part of this data source is the measurement supplied by the ANALYZE/DISK utility. This measurement set contains one record for each file in each directory on each disk included in the analysis. This data is the basis for the CA MICS DESDKU file and also supplies some information for the DESDKD file.

By comparison, the other measurements recorded by the DEXSUS routine account for a relatively small volume of data.

6.1.3 - Transporting Data from VMS to MVS

Because CA MICS is run under MVS and the VMS operating system is a separate entity, the VMS data must be transported to MVS. There are numerous ways to get the VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS data to MVS. The least desirable method would be to dump these files to tape on the VAX system and move them to the MVS system for processing.

You can use FileExchange, a CA software product, to transport files between your VAX/VMS systems and your MVS systems.

The DECnet/SNA Data Transfer Facility (DTF), a software product from Compaq Computer Corporation, also allows you to transmit data from your VMS systems to your MVS system.

Another alternative is FTP between VMS and MVS. If you choose this method, be sure to use versions of FTP that support record structure transfers of binary data in block mode. This is required because the logical record structure presented to the Analyzer must be the same as the original data records. In other words, use an FTP that will preserve the variable length record layout from VMS across to MVS.

For data transfer by tape, there are a number of public domain tape utility programs for VMS systems that convert VMS files to MVS variable block format. These are generally available through user groups, and are typically classified as "foreign tape" utility programs.

Data Format Processed by the VAX/VMS Analyzer

Whatever method you employ for getting the VMS data to CA MICS, be sure to obtain and transport the recording files in their original binary format and with a record format of variable blocked (RECFM=VB,LRECL=32752,BLKSIZE=32756). This is the format expected by the CA MICS VAX/VMS Analyzer input routines. Use of a maximum size record length and block size ensures that data records will not be truncated.

6.1.4 - Errors Identified in Measurement Data

1. VMS Release 4.7 errors.

   - SCS record does not always reset counters to 0 for the next record; this can affect all fields in the DEMSCS file.

2. VMS Release 5.0 and above.

   - The disk I/O queue length (DSKIOQUE) in an MP environment is not always reset to 0 at the start of a measurement interval. This can cause a queue length of some number with an I/O rate (DSKPSIO) of 0. When this occurs, the response time (DSKAVRES) that is calculated from these values is set to missing.

   - One field from the POOL record has garbage data stored on a sporadic but frequent basis. These fields are the count of large request packets. The garbage number is always the same (2147483649). This problem was found to be the method that Compaq uses for this measurement. If the number is larger than 499, then VMS MONITOR stores this large number.

   - If you have a TK50 or TK70 model tape drive, then the nodename in the SCS record will be blank. Compaq is aware of this problem.

3. VMS Release 5.4

   - Refer to the "VMS Version 5.4 Release Notes" published by Compaq Computer Corporation for additional information on problems and corrections in VMS. Chapters 2, 3, and 5 of this document contain specific references for VMS MONITOR and ACCOUNTING.

6.2 - VMS ACCOUNTING Record Descriptions

The VMS ACCOUNTING recording file is a VAX RMS sequential file with variable length records. Each record in the file begins with a one-byte type field. The remaining fields are different in length and format for each record type.

The following sections describe the VMS ACCOUNTING records processed by the CA MICS VAX/VMS Analyzer:

1 - PROCESS TERMINATION Record
2 - IMAGE TERMINATION Record
3 - SYSTEM INITIALIZATION Record
4 - LOGIN FAILED Record
5 - PRINT QUEUED Record

6.2.1 - PROCESS TERMINATION Record

PROCESS accounting writes a PROCESS TERMINATION record whenever a process terminates. For example, a process is created for a user when the user logs on to the system. The user may then execute editors, compilers, linkers, system utilities, and previously developed application programs. In doing so, only one process has been created. With PROCESS accounting, when the user logs off, a single PROCESS TERMINATION record will be created.

The PROCESS TERMINATION record contains data describing a process that has been executing. Its record type is 1.

The PROCESS TERMINATION record consists of two packets:

- Identification Packet - Contains identification data such as process identification, member and group IDs, user name, account, nodename, terminal, jobname, node address, remote ID, and other related data.

- Resource Packet - Contains system usage data such as status, image count, CPU time, page faults, fault I/Os, working set peak, page file, direct I/O, buffered I/O, and volumes mounted.

Data from this record is written to the DEAPRC file.

6.2.2 - IMAGE TERMINATION Record

IMAGE accounting writes an IMAGE TERMINATION record whenever an image executing within a process terminates. For example, a process is created for a user when the user logs on to the system. The user may then execute editors, compilers, linkers, system utilities, and previously developed application programs. In doing so, only one process has been created.

With PROCESS accounting, when the user logs off, a single PROCESS TERMINATION record will be created. With IMAGE accounting enabled, an IMAGE TERMINATION record will be created at the termination of each edit, compile, link, utility, or other program execution. It is clear that IMAGE accounting will create substantially more records than will PROCESS accounting. IMAGE accounting is useful when tracking the resource use of particular program executions is desired.

The IMAGE TERMINATION record contains data describing an image that had been executing. Its record type is 3.

The IMAGE TERMINATION record consists of three packets:

- Identification Packet - Contains identification data such as process identification, member and group IDs, user name, account, nodename, terminal, jobname, node address, remote ID, and other related data.

- Resource Packet - Contains system usage data such as status, image count, CPU time, page faults, fault I/Os, working set peak, page file, direct I/O, buffered I/O, and volumes mounted.

- Image Name Packet - Contains the name of the image run by the identified process.

Data from this record is written to the DEAIMG file.

6.2.3 - SYSTEM INITIALIZATION Record

The SYSTEM INITIALIZATION record is written whenever the VAX system is initialized. It is useful for measuring system up-time and system availability.

The SYSTEM INITIALIZATION record contains data describing the system initialization. Its record type is 5.

The SYSTEM INITIALIZATION record consists of two packets:

- Identification Packet - Contains identification data such as process identification, member and group IDs, user name, account, nodename, terminal, jobname, node address, remote ID, and other related data.

- Resource Packet - Contains system usage data such as status, image count, CPU time, page faults, fault I/Os, working set peak, page file, direct I/O, buffered I/O, and volumes mounted.

Data from this record is written to the DEAINT file.

6.2.4 - LOGIN FAILED Record

The LOGIN FAILED record contains data describing a LOGIN attempt that failed. It can be very useful for detecting security violations.

A LOGIN attempt can fail for a number of reasons: an incorrect user identification, an incorrect password, an attempt to use restricted privileges such as access at a restricted time of the day, an attempt to access prohibited system resources, or the maximum number of users has already logged on. Its record type is 7.

The LOGIN FAILED record consists of two packets:

- o Identification Packet - Contains identification data such as process identification, member and group IDs, user name, account, nodename, terminal, jobname, node address, remote ID, and other related data.

- o Resource Packet - Contains system usage data such as status, image count, CPU time, page faults, fault I/Os, working set peak, page file, direct I/O, buffered I/O, and volumes mounted.

Data from this record is written to the DEA_LF file.

6.2.5 - PRINT QUEUED Record

The PRINT QUEUED record contains data describing a print job that has been completed. Its record type is 8.

The PRINT QUEUED record consists of two packets:

- o Identification Packet - Contains identification data such as process identification, member and group IDs, user name, account, nodename, terminal, jobname, node address, remote ID, and other related data.

- o Print Resource Packet - Contains print operation data such as job status, time the job was queued, time the job was started, CPU time used by the print symbiont, pages printed, queue I/O (QIO) count, and GET count.

Data from this record is written to the DEAPRQ file.

6.3 - VMS MONITOR Record Descriptions

The VMS MONITOR recording file is a VAX RMS sequential file with variable length records. Each record in the file begins with a one-byte type field. The remaining fields are different in length and format for each record type.

The following sections describe the VMS MONITOR records processed by the CA MICS VAX/VMS Analyzer:

1 - FILE HEADER Record
2 - SYSTEM INFORMATION Record
3 - CLUSTER Record
4 - DECNET Record
5 - DISK Record
6 - DLOCK Record
7 - FCP Record
8 - FILE SYSTEM CACHE Record
9 - IO Record
10 - LOCK Record
11 - MODES Record
12 - MSCP SERVER Record
13 - PAGE Record
14 - POOL Record
15 - PROCESSES Record
16 - RMS Record
17 - SCS Record
18 - STATES Record
19 - SYSTEM Record
20 - TRANSACTION Record
21 - VECTOR Record

6.3.1 - FILE HEADER Record

The FILE HEADER record is the first record in every VMS MONITOR file. Its record type is 128. It contains the interval in seconds between VMS MONITOR data collections, the version of VMS MONITOR collecting the data, and other miscellaneous data.

6.3.2 - SYSTEM INFORMATION Record

The SYSTEM INFORMATION record is the second record in every VMS MONITOR file. Its record type is 129. It contains the time the system was booted, the number of CPUs in the system, the name of this node, and other miscellaneous data.

6.3.3 - CLUSTER Record

The CLUSTER record contains data describing clusterwide CPU, memory, and locking activity. Its record type is 19. Note that when CLUSTER records are requested, DISK and MODES records are also recorded even if not explicitly requested.

This record contains data such as CPU busy, free list size, and a number of lock request counters.

Data from this record is written to the DEMSPR file.

6.3.4 - DECNET Record

The DECNET record contains data describing the operation of the DECnet-VAX subsystem. Its record type is 8.

This record contains data such as local packets arriving and departing, packets lost, receiver buffer failures, and other miscellaneous data.

Data from this record is written to the DEMSPR file.

6.3.5 - DISK Record

The DISK record contains data describing all disk devices in the system. Its record type is 12.

This record contains data such as the disk controller, unit number, cluster node, volume name, count of I/O operations, and I/O request queue length.

Data from this record is written to the DEMDSK file.

6.3.6 - DLOCK Record

The DLOCK record contains data describing the operation of the Distributed Lock Management Facility. Its record type is 14.

This record contains data on new locks, lock conversions, unlocks, blocking asynchronous system traps (ASTs), and other data elements.

Data from this record is written to the DEMSPR file.

6.3.7 - FCP Record

The FCP record contains data describing the operation of the file system ACPs. Its record type is 5.

This record contains data on FCP calls, disk allocations, new files created, read and write I/Os, CPU time used, page faults, window turns, files opened, and other data elements.

Data from this record is written to the DEMSPR file.

6.3.8 - FILE SYSTEM CACHE Record

The FILE SYSTEM CACHE record contains data describing the operation of the caches for the ACPs and XQPs. Its record type is 11.

This record contains data on cache hits and attempts for directory file control blocks (FCBs), directory data, file headers, file IDs, extents, quota caches, and storage bitmaps.

Data from this record is written to the DEMSPR file.

6.3.9 - IO Record

The IO record contains data describing the operation of the I/O subsystem. Its record type is 4.

This record contains data on direct I/Os, buffered I/Os, files opened, page reads, read I/Os, page writes, write I/Os, inswaps, free page count, modified page count, and other related data elements.

Data from this record is written to the DEMSPR file.

6.3.10 - LOCK Record

The LOCK record contains data describing the operation of the VMS Lock Manager. Its record type is 7.

This record contains data on enqueues, dequeues, blocking ASTs, deadlocks, current locks, and current resources.

Data from this record is written to the DEMSPR file.

6.3.11 - MODES Record

The MODES record contains data describing the time spent in each of the processor modes. Its record type is 2. A set of data is provided for each CPU in the system.

This record contains CPU time data for the following modes: interrupt, synchronization, kernel, executive, supervisor, user, compatibility, and idle.

Data from this record is written to the DEMMOD file.

6.3.12 - MSCP SERVER Record

The MSCP SERVER record contains data describing the activities of the MSCP server. Its record type is 21.

This record contains data describing the MSCP requests, reads, writes, fragments, splits, buffer waits, and block I/Os.

Data from this record is written to the DEMSPR file.

6.3.13 - PAGE Record

The PAGE record contains data describing the operation of the Page Management System. Its record type is 3.

This record contains data describing page faults, reads, read I/O, writes, write I/O, page faults, free page count, and modified page count.

Data from this record is written to the DEMSPR file.

6.3.14 - POOL Record

The POOL record contains data describing space allocation in the nonpaged dynamic pool. Its record type is 6.

This record contains data describing the free and in use status of small, intermediate, and large packets, largest and smallest blocks, and unused and used pool space.

Data from this record is written to the DEMSPR file.

6.3.15 - PROCESSES Record

The PROCESSES record contains data describing all processes in the system. Its record type is 0.

This record contains data for each executing process: global page count, process page count, status, direct I/Os, page faults, CPU time, buffered I/Os, and other related data.

Data from this record is written to the DEMPRO file.

6.3.16 - RMS Record

The RMS record contains data describing the VMS Record Management Services for specified files. Its record type is 20.

This record contains data for each selected file: file organization, GETs, PUTs, updates, deletes, reads, writes, and many other related data elements.

Data from this record is written to the DEMRMS file.

6.3.17 - SCS Record

The SCS record contains data describing the VMS System Communication Services for specified files. Its record type is 15.

This record contains data for each node: datagrams sent, received, and discarded; sequenced messages sent and received; block transfers and kilobytes transmitted; and other related data.

Data from this record is written to the DEMSCS file.

6.3.18 - STATES Record

The STATES record contains data describing the number of processes in each of the scheduler states. Its record type is 1.

This record contains data for each scheduler state: collided page wait, miscellaneous resource wait, common event flag, page fault wait, local event flag, hibernate, suspended, free page wait, and compute state.

Data from this record is written to the DEMSPR file.

6.3.19 - SYSTEM Record

The SYSTEM record contains data describing the overall operation of the three major system components: CPU, memory, and I/O. Its record type is 17.

This record contains data regarding CPU busy, process count, page faults, read I/Os, free page count, modified page count, direct I/Os, and buffered I/Os.

Data from this record is written to the DEMSPR file.

6.3.20 - TRANSACTION Record

The TRANSACTION record contains data describing transaction processing statistics for DECdtm services. Its record type is 22.

This record contains data on including transaction starts, prepares, commits, aborts, remote branches, and ends. Transaction duration distributions are also provided in this record.

Data from this record is written to the DEMSPR file.

6.3.21 - VECTOR Record

The VECTOR record contains data describing the scheduling of vector consumers on vector present processors. Its record type is 23.

This record contains the CPU Identifier and the CPU time of Vector Consumers.

Data from this record is written to the DEMMOD file.

6.4 - CA MICS VMS System Usage (DEXSUS) Record Descriptions

The CA MICS VMS System Usage (DEXSUS) recording file, named DEXSUS.DAT, is a VAX RMS sequential file with variable length records. There are two types of records in the file: separators and measurements. Separators begin with the string "DEXSUS-" followed by a separator-type identifier. Separators may also be suffixed with special measurement values. Measurement records are placed between separator records, and constitute sets of data collected from VMS utility programs or other VMS commands. For some types of measurement records, such as ANALYZE/DISK, each record in the measurement set begins with a one-byte type field. In general, the records in the DEXSUS recording file are all different in length and format for each separator and measurement record type.

The CA MICS VMS System Usage (DEXSUS) records are a combination of fields and records from several VMS Utilities supported by Compaq Computer Corporation, including:

- VMS ANALYZE/DISK_STRUCTURE Utility
- VMS SYSMAN Utility, specifically the DISKQUOTA SHOW command
- VMS SHOW command
- VMS DCL lexical functions

The following sections describe the CA MICS VMS System Usage (DEXSUS) records processed by the CA MICS VAX/VMS Analyzer:

1 - DEXSUS FILE HEADER Record
2 - DEXSUS SYSTEM INFORMATION Records
3 - DEXSUS SYI1 Record
4 - DEXSUS SYI2 Record
5 - DEXSUS SYI3 Record
6 - DEXSUS SYI4 Record
7 - DEXSUS SYSTEM MEMORY 1 Records
8 - DEXSUS SYSTEM MEMORY 2 Records
9 - DEXSUS DISK DEVICE Records
10 - DEXSUS DISK QUOTA Records
11 - DEXSUS DISK USAGE Records
12 - DEXSUS FILE TRAILER Record

6.4.1 - DEXSUS FILE HEADER Record

The DEXSUS FILE HEADER Record is the first record in every MICS DEXSUS file. The separator record type is 00. This record contains the DEXSUS start time. No measurement records are associated with this separator.

Separator record layout:

```
DEXSUS-00-HDR1-DEX4100-F$TIME()
```

6.4.2 - DEXSUS SYSTEM INFORMATION Records

The DEXSUS SYSTEM INFORMATION Records begin with a type 01 separator record. This record contains the logical name SYS$NODE. The measurement records following this separator are the output from the VMS command: SHOW SYSTEM.

Separator record layout:

```
DEXSUS-01-SYS1-F$TRNLNM("SYS$NODE")
```

6.4.3 - DEXSUS SYI1 Record

The DEXSUS SYI1 Record is a type 11 separator record. This record contains the hardware name from F$GETSYI. No measurement records are associated with this separator.

Separator record layout:

```
DEXSUS-11-SYI1-F$GETSYI("HW_NAME")
```

6.4.4 - DEXSUS SYI2 Record

The DEXSUS SYI2 Record is a type 12 separator record. This record contains the VMS version from F$GETSYI. No measurement records are associated with this separator.

Separator record layout:

```
DEXSUS-12-SYI2-F$GETSYI("VERSION")
```

6.4.5 - DEXSUS SYI3 Record

The DEXSUS SYI3 Record is a type 13 separator record. This record contains the VMS boot time from F$GETSYI. No measurement records are associated with this separator.

Separator record layout:

```
DEXSUS-13-SYI3-F$GETSYI("BOOTTIME")
```

6.4.6 - DEXSUS SYI4 Record

The DEXSUS SYI4 Record is a type 14 separator record. This record contains the available CPU count from F$GETSYI. No measurement records are associated with this separator.

Separator record layout:

```
DEXSUS-14-SYI4-F$GETSYI("AVAILCPU_CNT")
```

6.4.7 - DEXSUS SYSTEM MEMORY 1 Records

The DEXSUS SYSTEM MEMORY 1 Records begin with a type 20 separator record. The measurement records following this separator are the output from the VMS command: SHOW MEMORY/FULL/POOL/FILE.

Separator record layout:

```
DEXSUS-20-MEM1
```

6.4.8 - DEXSUS SYSTEM MEMORY 2 Records

The DEXSUS SYSTEM MEMORY 2 Records begin with a type 21 separator record. The measurement records following this separator are the output from the VMS command: SHOW MEMORY/PHYSICAL

Separator record layout:

```
DEXSUS-21-MEM2
```

6.4.9 - DEXSUS DISK DEVICE Records

The DEXSUS DISK DEVICE Records begin with a type 30 separator record. This record contains the device name and device class determined from F$GETDVI. The measurement records following this separator are the output from the VMS command: SHOW DEVICE/FULL.

Separator record layout:

```
DEXSUS-30-DEV1-DEV-CLAS
```

6.4.10 - DEXSUS DISK QUOTA Records

The DEXSUS DISK QUOTA Records begin with a type 32 separator record. This record contains the device name and device class determined from F$GETDVI. The measurement records following this separator are the output from the VMS SYSMAN Utility command: DISKQUOTA SHOW * /DEV=dev.

Separator record layout:

```
DEXSUS-32-DQU1-DEV-CLAS
```

6.4.11 - DEXSUS DISK USAGE Records

The DEXSUS DISK USAGE Records begin with a type 33 separator record. This record contains the device name and device class determined from F$GETDVI. The measurement records following this separator are the output from the VMS ANALYZE/DISK_STRUCTURE Utility. Two measurement record types from the ANALYZE/DISK_STRUCTURE Usage File are processed: the Identification record (type 1) and the File Summary record (type 2).

Separator record layout:

```
DEXSUS-33-ADU1-DEV-CLAS
```

6.4.12 - DEXSUS FILE TRAILER Record

The DEXSUS FILE TRAILER Record is the last record in every MICS DEXSUS file. The separator record type is 99. This record contains the DEXSUS end time. No measurement records are associated with this separator.

Separator record layout:

```
DEXSUS-99-EOF1-F$TIME()
```

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 7

DEFINING PARAMETERS

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 7

DEFINING PARAMETERS

Chapter 7. DEFINING PARAMETERS

This chapter describes how to define the parameters that are required for installing the CA MICS VAX/VMS Analyzer. The CA MICS system administrator should use this chapter as a detailed reference in conjunction with the PIOM.

To define CA MICS parameters, you must gain an understanding of your installation and its needs and translate that understanding into CA MICS parameters.

In particular, this chapter requests that you:

- o Arrive at a number of policy decisions.
- o Fill out several worksheets.
- o Translate the worksheet entries into the corresponding CA MICS parameter library member entries.

These activities represent the major portion of the product installation process.

This chapter focuses on considerations that are unique to the CA MICS VAX/VMS Analyzer. Chapters 2 and 3 of the CA MICS PIOM Guide document the mechanics of the CA MICS installation process and include checklists that describe each installation step.

If you have a question at any time during your review of the material presented here, please contact the CA MICS Product Support Group.

This chapter contains information on the following topics:

1 - Environmental Considerations
2 - Complex Level Parameters
3 - Unit Level Parameters
4 - Other Related Parameters

7.1 - Environmental Considerations

Before coding product parameters, you need to know about the VAX environment(s) at your installation. Before you specify parameters for the product:

- o Review the default options for the parameters to determine their applicability to your site.
- o Review reporting requirements to determine whether or not you need to code MICF inquiries to satisfy your reporting needs.
- o Review data base unit specifications to determine which unit or units should include VAX/VMS data.

7.2 - Complex Level Parameters

This section shows you how to specify the complex level parameters that define the processing of the CA MICS VAX/VMS Analyzer. The following topics are covered:

1 - Analyzer Definition Statements (DEXGENIN)
2 - VAX Account Code Structure (DEXACCT)
3 - Code the VAX Account Code Exit (DEXACRT)
4 - Code the VAX/VMS Disk Account Code Exit (DEXACRTD)

7.2.1 - Analyzer Definition Statements (DEXGENIN)

A generation definition statement member is provided for the CA MICS VAX/VMS Analyzer in sharedprefix.MICS.GENLIB(DEXGENIN).

Complex-level options are available with this product. All file tailoring must be done at the FILE and NAME statement level. See the System Modification Guide Sections 4 and 6 for discussions of these statements and file tailoring.

7.2.2 - VAX Account Code Structure (DEXACCT)

In CA MICS, data about VAX/VMS activity is stored by account numbers in the following files:

In the VAX/VMS Accounting Information Area files:

- o VMS ACCOUNTING Process Term File (DEAPRC)
- o VMS ACCOUNTING Image Term File (DEAIMG)
- o VMS ACCOUNTING Initialization File (DEAINT)
- o VMS ACCOUNTING LOGIN Failed File (DEA_LF)
- o VMS ACCOUNTING Print Queued File (DEAPRQ)

In the VMS System Usage Information Area file:

- o VMS Disk Usage File (DESDKU)

The parameters you specify in sharedprefix.MICS.PARMS(DEXACCT) define the number of account code fields that will be carried in the files listed above within the CA MICS database, the length of each field, and the SAS long names that are associated with each field. Note that once you have defined these fields, you must provide a routine that assigns them values (see Sections 7.2.3 and 7.2.4).

PREPARATION

Each installation has its own method for associating VAX/VMS work with the responsible projects or departments. Prior to defining the account code parameters required by CA MICS in this area, you should investigate your installation's accounting standards to:

- o Identify the coding system (e.g., cost center coding system identifying the division, department, project, and employee).
- o Identify how the codes are specified.
- o Identify if and how the codes are verified to ensure that they correspond to a valid definition. We recommend that account code validation be performed in all cases and that unidentified or invalid account codes be assigned to a special installation overhead account code (see Sections 7.2.3 and 7.2.4). This approach provides two benefits. It filters out invalid codes from inclusion in the database and, therefore, requires less DASD space to hold the data in the files containing account code elements. Also, it makes it easy to see how much of this unidentifiable activity is taking place.

Once you have determined the accounting structure, consider the following when determining the number of account fields (CA MICS supports from one to nine) required to meet your installation's reporting and analysis needs:

- o Account fields are part of the file keys for the files that support them. As such, at least one record is generated for each combination of values. (More than one record may be generated because other fields also make up the file keys.) The trade-off you must make is between keeping a fine level of detail via the account code fields and, therefore, having a large database, and keeping a small level of detail and perhaps not meeting your reporting or analysis requirements.
- o If you anticipate needing to expand the account code structure in the future, establish an extra account code now to eliminate the need to retrofit the database to add the new account code.

VAX/VMS account code fields names are in the form "DEXACTn", where 'n' is the sequential number of the account code field. If three fields are defined, they will be DEXACT1, DEXACT2, and DEXACT3 (in your accounting structure, these might identify division, department, and project, respectively). The sequential number is called the account code field "level" number. There is a maximum of nine levels. A sample VAX/VMS account code structure is provided in sharedprefix.MICS.PARMS(DEXACCT).

Figure 7-1 provides a worksheet for collecting the data. Once you have completed this form, code the contents of sharedprefix.MICS.PARMS(DEXACCT) as follows:

- o A separate statement is coded for each account code level.
- o Blank statements are permitted. Comments are coded by beginning the statement with an '*'.
- o Account levels are provided in order, starting with "1".
- o Up to nine levels are permitted with no gaps between the numbers permitted.
- o The statement format is freeform but positional. All parameters are required. The form of each statement is as follows:

```
n m 'SAS long name to be assigned to DEXACTn'
```

The parameters to be coded are:

ACCOUNT LEVEL n

The level of importance of each element is specified with level 1 being the most important (major field) and the highest level number being the least important (minor field). From one to nine levels can be specified.

FIELD LENGTH m

The length of the specified element. The length may be in the range from 1 to 30.

ACCOUNT CODE LEVEL DESCRIPTION

The title that describes the account code field. The length of the title is limited to 40 characters.

A sample DEXACCT member with three levels is illustrated below:

```
1  3 'DIVISION'
2  6 'UIC GROUP'
3 12 'USER NAME'
```

```
+-------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET: VAX/VMS Account Code Level Definition|
|                                                                         |
| PARMS Library Member is DEXACCT                                         |
| Reference Sections: 7.2.2                                               |
+-------------------------------------------------------------------------+
|                                                                         |
|  ACCOUNT    FIELD                                                       |
| CODE LEVEL  LENGTH  ACCOUNT CODE LEVEL DESCRIPTION                      |
|   (1-9)                                                                 |
|                                                                         |
|     _         __    '_________________________________________'        |
|                                                                         |
|     _         __    '_________________________________________'        |
|                                                                         |
|     _         __    '_________________________________________'        |
|                                                                         |
|     _         __    '_________________________________________'        |
|                                                                         |
|     _         __    '_________________________________________'        |
|                                                                         |
|     _         __    '_________________________________________'        |
|                                                                         |
|     _         __    '_________________________________________'        |
|                                                                         |
|     _         __    '_________________________________________'        |
|                                                                         |
|     _         __    '_________________________________________'        |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
+-------------------------------------------------------------------------+
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-------------------------------------------------------------------------+
```

Figure 7-1. VAX/VMS Account Code Level Definition Worksheet

7.2.3 - Code the VAX/VMS Account Code Exit (DEXACRT)

The account code derivation routine is a user-written routine located in sharedprefix.MICS.PARMS(DEXACRT). It is invoked for each account record processed. This routine builds the account code data elements for the VAX/VMS Accounting Information Area files listed in section 7.2.2. For example, if the DEXACCT member in the sharedprefix.MICS.PARMS library defines four account code levels, this user routine has the responsibility of building the data value of those four fields for each observation processed.

The account code routine is written in the SAS language. The testing and accuracy of the process is the user's responsibility; however, the CA MICS VAX/VMS Analyzer provides a sample account code exit routine to help verify the accuracy of the user modifications to the routine. The worksheet for structuring the VAX/VMS account code derivation routine is shown in Figure 7-2.

The objective of the account code derivation process is to build the data elements DEXACT1 to DEXACT9 (as many as have been defined in DEXACCT) from the account detail data.

To limit retention of a DEXACTn element at a particular time-span, the sharedprefix.MICS.PARMS(DEXACCT) member must define that account code with the TIMESPAN mask, T(xxxxx). This is described in the CA MICS PIOM Guide, Section 2.3.1.7.

A sample DEXACRT member is illustrated below:

```
 /* DEXACRT - DEX ACCOUNT CODE EXIT ROUTINE

    THIS ROUTINE IS A USER CODED ROUTINE TO BUILD
    THE ACCOUNT FIELDS DEFINED BY THE 'DEXACCT'
    MEMBER IN MICS.PARMS. THIS ROUTINE MUST CORRECTLY
    BUILD EACH OF THE DEFINED 'DEXACT'  CODES FROM
    1 TO N.

    THIS ROUTINE MUST BE CODED ACCORDING TO THE
    FOLLOWING RULES:

        1.  PRIMARY DATA ELEMENTS AVAILABLE FOR
            THIS ROUTINE INCLUDE:

     DEXPID   - EXT. PROCESS ID   DEXTERM  -  TERMINAL
     DEXOWNID - OWNER ID          DEXJOB   -  JOBNAME
     DEXGRP   - UIC GROUP CODE    DEXJOBID -  JOB ID
     DEXMBR   - UIC MEMBER CODE   DEXQUEUE -  QUEUE
     DEXPRIV1 - PRIVILEGE 1       TYPE     -  RECORD TYPE
     DEXPRIV2 - PRIVILEGE 2       PRXTYPE  -  PRCTYPE,IMGTYPE
     DEXUSER  - USERNAME
     DEXACCNT - ACCOUNT
     DEXNODE  - NODE NAME OF REMOTE PROCESS

      THESE ELEMENTS ARE AVAILABLE FROM THE IDENTIFICATION
      PACKET OF EACH ACCOUNTING RECORD TYPE.
   ******************************************************** */

/* ******************************************************
   SAMPLE ACCOUNT CODE DERIVATION EXIT
   ******************************************************** */

 IF DEXACCNT NE ' ' THEN DO;
   DEXACT1 = SUBSTR(DEXACCNT,1,3);
   DEXACT2 = PUT(DEXGRP,6.);
   DEXACT3 = DEXUSER;
 END;
 ELSE DO;
  /* DEFAULT ACCOUNT CODES*/
   DEXACT1 = 'OVHD';
   DEXACT2 = '***';
   DEXACT3 = '***';
 END;
  /* END OF ROUTINE     */
```

```
INSTALLATION PREPARATION WORKSHEET: VAX/VMS Account Code Routine Defined

PARMS Library Member is DEXACRT
Reference Sections: 7.2.3

* VALIDATE FOR VALID ACCOUNT CODES, WHERE POSSIBLE ;

IF { account data is valid } THEN DO;

* BUILD ACCOUNT CODE FIELDS AS IN THE WORKSHEET 7-1

    DEXACT1=field source 1 ;
    ...
    ...
    ...
    DEXACTn=field source n ;
    ________________________________________________________________________
    ________________________________________________________________________
    ________________________________________________________________________
    ________________________________________________________________________
    ________________________________________________________________________

END;
ELSE DO;

* ROUTINE TO BUILD INSTALLATION OVERHEAD ACCOUNT CODES ;

    DEXACT1='overhead category' ;
    ...
    ...
    ...
    DEXACTn='overhead category' ;
    ________________________________________________________________________
    ________________________________________________________________________
    ________________________________________________________________________
    ________________________________________________________________________

END;

....5...10...15...20...25...30...35...40...45...50...55...60...65...70..
```

Figure 7-2. VAX/VMS Account Code Routine Definition Worksheet

7.2.4 - Code the VAX/VMS Disk Account Code Exit (DEXACRTD)

The disk account code derivation routine is a user-written routine located in sharedprefix.MICS.PARMS(DEXACRTD). It is invoked for each disk usage accounting record processed. This routine builds the account code data elements for the VMS System Usage Information Area files listed in section 7.2.2. For example, if the DEXACCT member in the sharedprefix.MICS.PARMS library defines four account code levels, this user routine has the responsibility of building the data value of those four fields for each observation processed.

The account code routine is written in the SAS language. The testing and accuracy of the process is the user's responsibility; however, the CA MICS VAX/VMS Analyzer provides a sample account code exit routine to help verify the accuracy of the user modifications to the routine. The worksheet for structuring the VAX/VMS account code derivation routine is shown in Figure 7-2a.

The objective of the disk account code derivation process is to build the data elements DEXACT1 to DEXACT9 (as many as have been defined in DEXACCT) from the account detail data.

To limit retention of a DEXACTn element at a particular time-span, the sharedprefix.MICS.PARMS(DEXACCT) member must define that account code with the TIMESPAN mask, T(xxxxx). This is described in the CA MICS PIOM Guide, Section 2.3.1.7.

A sample DEXACRTD member is illustrated below:

```
/* DEXACRTD - DEX DISK USAGE ACCOUNT CODE EXIT ROUTINE

   THIS ROUTINE IS A USER CODED ROUTINE TO BUILD
   THE ACCOUNT FIELDS DEFINED BY THE 'DEXACCT'
   MEMBER IN MICS.PARMS. THIS ROUTINE MUST CORRECTLY
   BUILD EACH OF THE DEFINED 'DEXACT'  CODES FROM
   1 TO N, FOR THE DESDKU FILE.

   THIS ROUTINE MUST BE CODED ACCORDING TO THE
   FOLLOWING RULES:

      1.  PRIMARY DATA ELEMENTS AVAILABLE FOR
          THIS ROUTINE INCLUDE:

          DEXGRP   - UIC GROUP CODE
          DEXMBR   - UIC MEMBER CODE
          DEXNODE  - NODE NAME WHERE MEASUREMENT WAS TAKEN
          DEXDEVIC - DISK DEVICE NAME
          DKUDIRTL - DIRECTORY TOP LEVEL
          DKUDIR   - DIRECTORY NAME
          DKUFILE  - FILE NAME

       THESE ELEMENTS ARE AVAILABLE FROM THE DEXSUSA MODULE
       PROCESSING FOR THE DESDKU FILE.
  ******************************************************* */

SELECT(DEXAFILE);
WHEN ('DESDKU') DO;

/* ***************************************************** */
/* SAMPLE DISK ACCOUNT CODE DERIVATION EXIT              */
/* ***************************************************** */

IF DEXNODE  NE ' ' THEN DO;
  DEXACT1 = SUBSTR(DEXNODE,1,3);
  DEXACT2 = PUT(DEXGRP,6.);
  DEXACT3 = DKUDIRTL;
END;
ELSE DO;
  /* DEFAULT ACCOUNT CODES*/
  DEXACT1 = 'OVH';
  DEXACT2 = '***';
  DEXACT3 = '***';
END;

/* ***************************************************** */
/* END OF ROUTINE                                        */
/* ***************************************************** */
```

```
  END; /* END OF WHEN */
  OTHERWISE;
  END; /* END OF SELECT */
```

```
INSTALLATION PREPARATION WORKSHEET: VAX/VMS Disk Account Code Routine

PARMS Library Member is DEXACRTD
Reference Sections: 7.2.4

* VALIDATE FOR VALID ACCOUNT CODES, WHERE POSSIBLE ;

IF { account data is valid } THEN DO;

* BUILD ACCOUNT CODE FIELDS AS IN THE WORKSHEET 7-1

    DEXACT1=field source 1 ;
    ...
    ...
    ...
    DEXACTn=field source n ;
    ______________________________________________________________________
    ______________________________________________________________________
    ______________________________________________________________________
    ______________________________________________________________________
    ______________________________________________________________________

END;
ELSE DO;

* ROUTINE TO BUILD INSTALLATION OVERHEAD ACCOUNT CODES ;

    DEXACT1='overhead category' ;
    ...
    ...
    ...
    DEXACTn='overhead category' ;
    ______________________________________________________________________
    ______________________________________________________________________
    ______________________________________________________________________
    ______________________________________________________________________

END;

....5...10...15...20...25...30...35...40...45...50...55...60...65...70..
```

Figure 7-2a. VAX/VMS Disk Account Code Routine Definition Worksheet

7.3 - Unit Level Parameters

This section contains information on the unit level parameters for the CA MICS VAX/VMS Analyzer. The following topics are covered:

1 - VAX/VMS Processing Options (DEXOPS)
2 - Input DD Statements for VAX/VMS (INPUTDEX)
3 - Input Needed to Generate VAX System Code (DEXPGEN)
4 - Database Space Modeling (DBMODEL)

7.3.1 - VAX/VMS Processing Options (DEXOPS)

This section shows you how to specify the operational statements that control CA MICS VSE/VMS Analyzer processing.

Operational statements are stored in the prefix.MICS.PARMS cccOPS member, where ccc is the component identifier, and are incorporated into the CA MICS system by running the prefix.MICS.CNTL(cccPGEN) job.

```
************************************************************
*                                                          *
*  NOTE:  CHANGES to prefix.MICS.PARMS(cccOPS) members     *
*         REQUIRE EXECUTION of prefix.MICS.CNTL(cccPGEN)   *
*         to take effect.                                  *
*                                                          *
*         In addition, any change to parameters that       *
*         impact the DAILY operational job JCL such as,    *
*                                                          *
*         o  changing RESTART NO to RESTART YES,           *
*                                                          *
*         o  WORK parameter changes when RESTART NO is in  *
*            effect,                                       *
*                                                          *
*         o  Specifying TAPEfff (if this product supports  *
*            a DETAIL level TAPE option),                  *
*                                                          *
*         o  or changes to prefix.MICS.PARMS(INPUTccc),    *
*                                                          *
*         will require regeneration of the DAILY job by    *
*         executing prefix.MICS.CNTL(JCLGEND) or by        *
*         specifying DAILY in prefix.MICS.PARMS(JCLGENU)   *
*         and executing prefix.MICS.CNTL(JCLGENU).         *
*                                                          *
*         Refer to the checklist (if provided) for updating *
*         cccOPS parameters and running required generation *
*         jobs.                                            *
*                                                          *
************************************************************
```

To define the VAX/VMS input data to CA MICS at the unit level, you code parameters in the prefix.MICS.PARMS(DEXOPS). To process data from multiple VAX/VMS systems in a single unit, you must specify each system that you want the unit to process.

The following are valid statements:

o OPTIONS - Specify one of these for every VAX/VMS node you want to process.

o MONITOR - Specify one of these for every VMS MONITOR input

```
              data file you want to process.

o ACCOUNT  - Specify one of these for every VMS ACCOUNTING
             input data file you want to process.

o DEXSUS   - Specify one of these for every VMS System Usage
             (DEXSUS) input data file you want to process.

o VERIFY   - Only one of these exists in DEXOPS.

o DISKACCT - (Optional) Only one of these exists in DEXOPS.

o COMMON   - (Optional) Only one of these for each datatype
             in DEXOPS. That is, one for MONITOR, one for
             ACCOUNT, and one for DEXSUS.

o WORK     - (Optional) n data_set_allocation_parameters

o MULT/NOMULT - (Optional) fff fff ... fff

o RESTART  - (Optional) YES/NO

o INCRUPDATE - Incremental Update options (optional)
```

OPTIONS STATEMENT

The OPTIONS statement associates a VAX nodename with an ORGSYSID.

```
+-------------------------------------------------------+
| OPTIONS orgsysid nodename                             |
+-------------------------------------------------------+
```

where:

'orgsysid' is the original SYSID used in the SYSID member of prefix.MICS.PARMS to identify this VAX system.

'nodename' is the VAX nodename that is used to run this VMS system. It is specified as a character string up to 10 characters in length. The first 10 characters of the VAX nodename in the VMS MONITOR type 129 record will be compared to this value during DAY074 processing.

Sample OPTIONS statement:

```
OPTIONS VMS1 FRED1
```

MONITOR STATEMENT

The MONITOR statement identifies the JCL reference to be used when reading VMS MONITOR data for this system.

```
+-------------------------------------------------------------+
| MONITOR orgsysid ddname                                     |
+-------------------------------------------------------------+
```

where:

'orgsysid' is the original SYSID specified on a previous OPTIONS statement.

'ddname' is the DDname that will be used to read VMS MONITOR data for this system. Specify 'ddname' as '*' if a COMMON statement provides the DDNAME for this data.

Sample MONITOR statement:

```
MONITOR VMS1 DEXMON01
```

ACCOUNT STATEMENT

The ACCOUNT statement identifies the JCL reference to be used when reading VMS ACCOUNTING data for this system.

```
+-------------------------------------------------------------+
| ACCOUNT orgsysid ddname                                     |
+-------------------------------------------------------------+
```

where:

'orgsysid' is the original SYSID specified on a previous OPTIONS statement.

'ddname' is the DDname that will be used to read VMS ACCOUNTING data for this system. Specify 'ddname' as '*' if a COMMON statement provides the DDNAME for this data.

Sample ACCOUNT statement:

```
ACCOUNT VMS1 DEXACC01
```

DEXSUS STATEMENT

The DEXSUS statement identifies the JCL reference to be used when reading DEXSUS data for this system.

```
+-------------------------------------------------------------+
| DEXSUS orgsysid ddname duration                             |
+-------------------------------------------------------------+
```

where:

'orgsysid' is the original SYSID specified on a previous OPTIONS statement.

'ddname' is the DDname that will be used to read DEXSUS data for this system. Specify 'ddname' as '*' if a COMMON statement provides the DDNAME for this data.

'duration' is the default duration to assume for disk file age when no prior record for the file can be located. For files previously processed, the value for duration is computed by the DEX input format routine from the current and previous sample times recorded by DEXSUS.

This value is expressed in hours, and is required on the DEXSUS statement for each system being processed. The file duration value is used to compute disk storage occupancy for VMS files in the DESDKU file and for each user ID code in the DESDKQ file.

Sample DEXSUS statement:

```
DEXSUS VMS1 DEXSUS01 12
```

VERIFY STATEMENT

The VERIFY statement directs the Analyzer to take the specified action when data from an input DDname specified on a MONITOR statement comes from a system not named on the associated OPTIONS statement for the ORGSYSID.

```
+-------------------------------------------------------------+
| VERIFY action                                               |
+-------------------------------------------------------------+
```

where:

'action' is ABORT or NOABORT. If the VERIFY statement is omitted, the default is NOABORT. When ABORT is coded and a mismatch is detected, a U998 abend is issued to halt processing. If the data is from a DDNAME specified on a COMMON MONITOR statement without the EXIT keyword, this option has no effect.

Sample VERIFY statement:

```
VERIFY ABORT
```

DISKACCT STATEMENT

The DISKACCT statement is used to specify your choice of DETAIL or DAYS level accounting support for VMS disk space usage processed from the DESDKU file. This statement is optional. If omitted, a default value of DETAIL will be assumed.

```
+-------------------------------------------------------+
| DISKACCT time-span                                    |
+-------------------------------------------------------+
```

where:

'time-span' is DETAIL or DAYS. This parameter specifies when, during the processing of the DESDKU file, to invoke the accounting routines generated by CA MICS Accounting and Chargeback.

You should consult with the people responsible for CA MICS Accounting and Chargeback before choosing a value for DISKACCT.

IMPORTANT
=========

You should ensure that the Component timespan field for the ACTJDK file in the CA MICS Accounting and Chargeback Journal File Eligibility Specifications (MWF;4;2;S;2) matches the value you specify for DISKACCT. This field is used to verify that the data elements used for the journal file are active in the specified time-span and are sequence elements.

DETAIL versus DAYS Level Accounting

DAYS results in DEX presenting a summarized observation to the accounting routine. The sequence/summary elements of the DESDKU file at the DAYS time-span are used to create a summarized observation that represents all DASD space occupied by that control break.

DAYS allows reasonable flexibility in qualification pricing in the accounting routine. For example, assume the sequence/summary elements of DESDKU at the DAYS level are:

```
SYSID  DEXDEVIC  DKUDIRTL  YEAR     MONTH
DAY    DEXACT1   DEXACT2   DEXACT3
```

CA MICS summarization routines will create a single observation for data sets whose values for these fields form a unique combination.

Accounting qualification techniques and algorithms allow for pricing of disk space by applying rates based on the actual values in these control or sequence elements in the above example.

Note that variables such as DEXGRP, DEXMBR, DKUDIR, and DKUFILE for are absent from the list. If you need to implement exception level pricing using character data elements not in the sort key of the DAYS time-span, consider coding DISKACCT DETAIL here in DEXOPS.

DETAIL causes DEX to invoke the accounting routine to price a data set at the DETAIL time-span. Each data set can be examined by the accounting cost algorithms and all data elements carried at the DETAIL time-span are available for inspection by the accounting code.

To continue with the example above, DKUDIR could be tested and a special rate could be applied to some files (DKUFILE) if they appeared in a given directory (DKUDIR).

It is important to note that DAYS-level DESDKU observations presented to the accounting code are derived from the DETAIL time-span of the DESDKU file. Regardless of the selection of DETAIL or DAYS, the disk account code assignment routine is always applied to the DETAIL level of DESDKU and, therefore, all detail-level variables are available to derive account codes. The account code elements are part of the set of sort/sequence variables for both DETAIL and DAYS levels.

Another important factor in choosing DETAIL versus DAYS is the effect that rounding will have on total charges. Because charges for very small files may round to zero, depending upon the rate charged, you may find the DAYS option preferable since the rounding will occur at a more summarized level. In other words, round-off may be reduced by using the DAYS option.

Since there are more observations at the DETAIL time-span than at the DAYS level, it follows that DETAIL will increase the space required for cycles of the accounting journal file (ACTJDK01). There is a direct correspondence between the number of observations on the DESDKU file and the accounting journal file that holds the charges developed from the DESDKU file.

Sample DISKACCT statement:

```
DISKACCT DAYS
```

COMMON STATEMENT

The COMMON statement is not required. It is designed primarily for handling VMS data that has been preprocessed into a special format for consolidated input of many systems through a single input DDNAME. If you do not have such a requirement, you should ignore this parameter.

The optional COMMON statement identifies the JCL reference to use when reading MONITOR, ACCOUNTING, or DEXSUS data for one or more VMS systems from the same input DDNAME. This technique normally requires special exit coding to determine the data type and system identification of each incoming record. Before using the COMMON statement, please refer to Chapter 9 for a complete discussion of input processing and exit routines. Section 9.4, Using COMMON Input DDnames and DEXINPUT Exits, describes the use of the COMMON statement in detail.

DEXOPS Example

The definition of the data items described above is made through the DEXOPS member in the prefix.MICS.PARMS library. A sample definition is illustrated below. Figure 7-3 provides a worksheet for collecting the data.

Sample definition:

```
VERIFY  ABORT
OPTIONS   VMS1  SYSTEMONE
  MONITOR VMS1  DEXMON01
  ACCOUNT VMS1  DEXACC01
  DEXSUS  VMS1  DEXSUS01  12
OPTIONS   VMS2  SYSTEMTWO
  MONITOR VMS2  DEXMON02
  ACCOUNT VMS2  DEXACC02
OPTIONS   VMS3  SYSTEM3
  ACCOUNT VMS3  DEXACC03
DISKACCT DAYS
```

WORK

This statement is optional. It enables sites experiencing either SAS WORK space allocation problems or out of work space conditions during DAYnnn or INCRnnn (where nnn is the job step number), daily or incremental update processing, to allocate multiple WORK files.

You can allocate multiple WORK files for use during the daily and/or incremental update job step. The maximum number of WORK files you can allocate varies by product. These additional work files are used in conjunction with the single work data set allocated by default using the JCLDEF parameters WORKUNIT and WORKSPACE.

Because the individual space allocation requirement for each WORK file is typically much smaller, it is more likely to be satisfied.

To take advantage of multiple WORK files support, edit prefix.MICS.PARMS(cccOPS) and insert a WORK statement as shown below:

```
WORK n data_set_allocation_parameters
```

where n is the number of WORK data sets

```
        NOTE:  The default is zero (0).
               The maximum is nine (9).

        data_set_allocation_parameters is one or more data
        set allocation parameters (for example, STORCLAS or
        SPACE) separated by spaces.
```

You can also specify the WORK parameter as the following:

```
WORK n XXX pppp ssss
```

where:

```
  n    is the number of WORK data sets
  XXX  is TRK or CYL
  pppp is the primary allocation
  ssss is the secondary allocation
```

Note: When allocating any number of SAS WORK data sets, be aware that one additional SAS WORK data set is automatically allocated to facilitate sorting. For example, if you allocate six SAS WORK data sets, you will actually get seven.

If you omit the data_set_allocation_parameters or the WORK parameter, the work data sets are allocated according to the values you specified for the WORKUNIT and WORKSPACE parameters in prefix.MICS.PARMS(JCLDEF). Use the data_set_allocation_parameters to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Note: If you allocate insufficient space for the WORK data sets, DAYnnn and/or INCRnnn processing will fail and can only be restarted from the beginning.

Note: If internal step restart is active, you can override the WORK data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the WORK keyword on the continuation line.

WORK accepts the engine/host options documented in the SAS Companion for the z/OS environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! Do not specify the DISP parameter.

Example 1:

```
WORK n STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss),RLSE)
```

where:

```
  n        - is the number of WORK data sets.
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to 8 characters.
  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.
  XXX      - is TRK or CYL.
  pppp     - is the primary allocation.
  ssss     - is the secondary allocation.
  RLSE     - specifies that free-space should be released
             when the data set is closed.
```

Example 2:

```
WORK n XXX pppp ssss
```

where:

```
  n        - is the number of WORK data sets.
  XXX      - is TRK or CYL.
  pppp     - is the primary allocation.
  ssss     - is the secondary allocation.

Example 3 (multiple lines):

WORK n STORCLAS=MICSTEMP UNIT=SYSDA
WORK   SPACE=(xxxx,(pppp,ssss),,,ROUND))

where:

  n        - is the number of WORK data sets.
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.
  UNIT     - specifies the generic unit for a new data set.
             The name can have up to eight characters.
  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.
  XXX      - is TRK or CYL.
  pppp     - is the primary allocation.
  ssss     - is the secondary allocation.

Note:  Since there is some performance impact when using
multiple WORK files, you should specify the minimum number of
WORK data sets to meet your work space requirements.  As a
start, try incrementing the number gradually beginning from
the default.


WORK Considerations
--------------------

How Much Space Should You Allocate?

o  First Time Implementation of Multiple Work Files

   If this is the first time you are implementing multiple
   work files for this product in this unit, review
   prefix.MICS.PARMS(JCLDEF) and find the WORKSPACE
   parameter.  It will resemble this sample statement:

   WORKSPACE      TRK 500 250

   The value shows the current SAS WORK space allocation for
   the unit as a single data set.  It also serves as the
   default value used in the unit's DAYnnn daily update
   (and/or INCRnnn incremental update) step unless you
   provide a WORK parameter.

   To achieve the equivalent work space allocation of
   WORKSPACE TRK 500 250 using multiple WORK data sets that
   will collectively share the work space requirements of
   the daily and/or incremental update step, you could code
   either one of these:

   WORK 2 SPACE=(TRK,(250,125))

   WORK 5 SPACE=(TRK,(100,50))

   To determine the total work space, multiply the number of
   WORK files (n) by the primary (pppp) and secondary (ssss)
   values specified.

   Note:  To simplify the example, only the SPACE parameter
   is shown above.  You can follow either with data set
   allocation parameters like UNIT or STORCLAS as required
   for your site.

o  Adjusting Allocation for Existing Multiple WORK Files

   If you have previously implemented multiple WORK file
   support for this product in this unit, and you want to
   change either the number of WORK files or the space
   allocations, examine prefix.MICS.PARMS(cccOPS) and find
   the existing WORK statement.

   -  If the existing WORK statement only specifies the
      number of WORK files but does not contain space
      allocation information as shown below:

      WORK 5

      Then each of the multiple WORK files is allocated
      using the values from the WORKSPACE parameter of
      prefix.MICS.PARMS(JCLDEF), as described earlier under
      First Time Implementation of Multiple Work Files.

      To increase workspace, you can increase the number of
      WORK files (for example, change WORK 5 to WORK 6,7,8,
      or 9), increase the space allocation in the WORKSPACE
      parameter, or do both.

      To decrease workspace, you can decrease the number of
      WORK files (for example, change WORK 5 to WORK 4,3,2,
      or 1), decrease the space allocation in the WORKSPACE
      parameter, or do both.

      You can also elect to explicitly specify the multiple
      WORK file space allocation by adding the space
      allocation values directly to the WORK statement.  This
      will remove the link to the prefix.MICS.PARMS(JCLDEF)
      WORKSPACE parameter for multiple WORK file space
      allocation.  This is recommended as it serves to
```

clearly document, in one place, how multiple WORK files are allocated.

- If the existing WORK statement does include space allocation as shown in the examples below:

  WORK 5 TRK 200 100

  or

  WORK 5 SPACE=(TRK,(200,100)) STORCLAS=MICSTEMP

  Simply change the values to meet your needs.

  If you need more work space, you can increase the number of WORK files (for example, change WORK 5 to WORK 6,7,8, or 9), increase the space allocation (for example, change TRK 200 100 to TRK 250 120), or do both.

  To decrease work space, you can decrease the number of WORK files (for example, change WORK 5 to WORK 4,3,2, or 1), decrease the space allocation (for example, change TRK 200 100 to TRK 150 80), or do both.

Note: If internal step restart is NOT active (RESTART NO) and you change the WORK parameter, you must:

o Run cccPGEN
o Run JCLGENU for DAILY (to regenerate DAILY) and, if incremental update is enabled, INCRccc

When internal step restart is active, (RESTART YES), then, when you change WORK and run cccPGEN, changes take effect immediately. There is no need to run JCLGENU.

SASWORK

This statement is optional.

The WORK DD statement in the CA MICS procedures allocates a temporary data set where SAS keeps its temporary data files and other items that SAS uses during processing of the current job.

By default, the space allocated is defined in the member prefix.MICS.PARMS(JCLDEF) with the WORKSPACE and WORKUNIT parameters, then generated into all the JCL procedures for a given unit.

With the SASWORK statement you have the option to override this unit-wide definition to specify the space allocation individually for the current step.

The format of the SASWORK statement is:

SASWORK data_set_allocation_parameters

where data_set_allocation_parameters is one or more data set allocation parameters (for example, STORCLAS or SPACE) separated by spaces.

You can also specify the SASWORK parameter as the following:

SASWORK XXX pppp ssss

where:

XXX is TRK or CYL
pppp is the primary allocation
ssss is the secondary allocation

If you omit the data_set_allocation_parameters or the SASWORK statement, the WORK data set is allocated according to the values you specified for the WORKUNIT and WORKSPACE parameters in prefix.MICS.PARMS(JCLDEF). Use the data_set_allocation_parameters to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the SASWORK keyword on the continuation line.

Example:

SASWORK STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss))

where:

STORCLAS - specifies a storage class for a new data set. The name can have up to 8 characters.
SPACE - specifies how much disk space to provide for a new data set being allocated.
XXX - is TRK or CYL.
pppp - is the primary allocation.
ssss - is the secondary allocation.

Note: If you change the SASWORK parameter, you must:

```
o  Run cccPGEN
o  Run JCLGENU for DAILY (to regenerate DAILY) and, if
   incremental update is enabled, INCRccc


MULTWORK|NOMULT fff fff ... fff
-------------------------------

Since multiple work files usage impacts performance, this
product provides these optional parameters so you can
restrict multiple work files usage to only those files having
excessive space requirements.

Note:  You can only use one of these optional parameters with
the WORK statement, NOT both.

The MULTWORK parameter restricts the use of multiple WORK
files to ONLY those listed after the MULTWORK keyword.

MULTWORK fff fff ... fff

where fff is the unique three character identifier

If you need multiple lines, repeat the MULTWORK on the
continuation line.

The NOMULT parameter forces the use of multiple WORK files
for all files EXCEPT those specified after the NOMULT
keyword.

NOMULT fff fff ... fff

where fff is the unique three character identifier

If you need multiple lines, repeat the NOMULT on the
continuation line.

The default is

 MULTWORK PRO STA MOD PGE IOV FCP POL LOK DNT FSC DSK DLK
 MULTWORK SCS SYT CLS RMS MCS SPR TRN VPR DKD DKQ DKU PRX SYU

 NOMULT   PRC IMG INT _LF PRQ

if neither MULTWORK or NOMULT parameters are specified.
```

```
RESTART YES/NO
--------------

This statement is optional.  Specify this to activate
internal step restart for this product's DAILY and/or INCRccc
database update job steps:

RESTART YES


If you do not specify or enable the RESTART parameter, then
this option defaults to the following and internal step
restart is disabled:

RESTART NO


************************************************************
*                                                          *
*  Note:  Changing the RESTART parameter (either from NO   *
*         to YES or from YES to NO) requires regeneration  *
*         of the DAILY operational job by executing        *
*         prefix.MICS.CNTL(JCLGEND) or by specifying       *
*         DAILY in prefix.MICS.PARMS(JCLGENU) and          *
*         executing prefix.MICS.CNTL(JCLGENU).             *
*                                                          *
*         If incremental update is active for this product, *
*         you must also regenerate the INCRccc job.        *
*                                                          *
************************************************************

Internal step restart can significantly reduce time and
resource usage to recover from daily and/or incremental
update processing failures.  CA MICS uses a
checkpoint/restart technique.

o  When internal step restart is activated, the database
   update job step "checkpoints" (or saves) intermediate
   results (work file contents) and the operational
   environment at the end of each processing phase.

o  Then, if required, the database update step can resume
   execution at the beginning of the processing phase in
   which the failure occurred.

o  Restart is accomplished by restoring the operational
   environment from the last checkpoint, bypassing completed
   processing phases, and resuming execution using
   intermediate results (work files) from the last
   checkpoint.

Note:  When you activate internal step restart (RESTART YES),
```

the following optional restart parameters are enabled. These parameters have no effect if restart is disabled (RESTART NO). For more details, see the individual parameter descriptions later in this section.

- o RESTARTCKPT data_set_allocation_parameters
- o RESTARTWORK data_set_allocation_parameters
- o DYNAMWAIT minutes

Processing Phases:

This product employs two database update processing phases followed by the two common roll-up phases.

Phase	Description
FORMAT	Read raw input data, convert to SAS format, and output to intermediate work files.
DBUPDATE	Sort intermediate work file contents, eliminate duplicate input data, prepare for DETAIL cycle creation, merge data across optional multiple work files, enhance data content, and create the new DETAIL cycle.
DYSUM	Summarize DETAIL data to create new DAYS cycles and to update current week-to-date and month-to-date cycles.
DYAGE	Cutover new database cycles to production and "age" existing cycles.

RESTART Considerations

- o Overhead

  Enabling internal step restart adds some overhead to the database update job step -- the cost of taking checkpoints and managing saved materials. Since this overhead is relatively constant and independent of input data volume, you may find that costs outweigh potential savings when input data volume is low, for example in a test unit. For high volume, production units, internal step restart support overhead should be a minor portion of total resource usage.

- o Cataloged Work Files

  When internal step restart is enabled, the SAS work data set, internal step restart control data set, and multiple work file data sets are allocated and cataloged with permanent dataset names so they will be retained for use in restart if the step abends. These data sets are deleted when the step completes successfully.

  Prior to enabling internal step restart support, these data sets were probably allocated on system "scratch" space with a temporary, system assigned data set names. If your installation standards do not allow "permanent" data sets on DASD volumes used for temporary work space, you may need to use the WORK, RESTARTCKPT, and RESTARTWORK parameters to direct the internal step restart data sets to a generic unit or storage class that allows cataloged data sets.

- o Dynamic Allocation

  When internal step restart is active, dynamic allocation is employed for the work data sets. If your installation restricts dynamic allocation of large, cataloged data sets, you may need to use the WORK, RESTARTCKPT, and RESTARTWORK parameters to direct work data set allocation to a generic unit or storage class where dynamic allocation is allowed.

- o Data Set Names

  The SAS work data set, internal step restart control data set, and multiple work file data sets are allocated and cataloged according to the standard CA MICS unit database data set name conventions. The default DDNAME and data set names are:

```
      o SAS work data set,
            //cccXWORK DD DSN=prefix.MICS.cccXWORK,.....

      o Internal step restart control data set,
            //cccXCKPT DD DSN=prefix.MICS.cccXCKPT,.....

      o Multiple work file data sets,
            //WORKnn   DD DSN=prefix.MICS.cccWRKnn,.....

    Since these data sets conform to the same data set name
    conventions as your existing CA MICS data sets, there
    should be few, if any, data set name related allocation
    issues.  However, it is possible to override the data set
    names if required.  Please contact CA MICS Product
    Support for assistance if you must alter data set names.


RESTARTCKPT
-----------

This statement is optional.  Specify the following to
override default data set allocation parameters for the
internal step restart checkpoint data set:

RESTARTCKPT   data_set_allocation_parameters

Note:  RESTARTCKPT is ignored when you specify RESTART NO.

The internal step restart checkpoint data set (or cccXCKPT
data set) contains processing status, control, and SAS
environmental information for internal step restart
processing checkpoints.  This includes a copy of the SAS WORK
format and macro catalogs, current macro variable values, and
a description of work files that may be needed to restart
DAYnnn processing.

By default, the cccXCKPT data set is allocated according to
the values you specified for the WORKUNIT and WORKSPACE
parameters in prefix.MICS.PARMS(JCLDEF).  Specify RESTARTCKPT
to override this default, either to alter the space
allocation or to use System Managed Storage (SMS) parameters
to control data set placement and characteristics.

Note:  If you allocate insufficient space for the cccXCKPT
data set, DAYnnn processing will fail and can only be
restarted from the beginning.

Note:  You can override the RESTARTCKPT data set allocation
parameters at execution-time using the //PARMOVRD facility.
For more information about execution-time override of dynamic
data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks,
according to SAS LIBNAME statement syntax.  If you need
multiple lines, repeat the RESTARTCKPT keyword on the
continuation line.

RESTARTCKPT accepts the engine/host options documented in the
SAS Companion for the z/OS Environment, including STORCLAS,
UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important!  DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

RESTARTCKPT  STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)

where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated, where:

             xxxx is TRK, CYL, or blklen
             pp is the primary allocation
             ss is the secondary allocation

             and ROUND specifies that the allocated space be
             "rounded" to a cylinder boundary when the unit
             specified was a block length.  ROUND is ignored
             with the TRK or CYL options.

Example 2 (multiple lines):

RESTARTCKPT  STORCLAS=MICSTEMP UNIT=SYSDA
RESTARTCKPT  SPACE=(xxxx,(pp,ss),,,ROUND)

where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  UNIT     - specifies the generic unit for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.


RESTARTWORK
-----------
```

This statement is optional. Specify the following to override default data set allocation parameters for the internal step restart WORK data set:

```
RESTARTWORK   data_set_allocation_parameters
```

Note: RESTARTWORK is ignored when you specify RESTART NO.

The internal step restart WORK data set (or cccXWORK data set) contains the intermediate work files that are not enabled to multiple work file support, including those files you may have specified on the optional NOMULT statement.

By default, the cccXWORK data set is allocated according to the values you specified for the WORKUNIT and WORKSPACE parameters in prefix.MICS.PARMS(JCLDEF). Specify RESTARTWORK to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Note: If you allocate insufficient space for the cccXWORK data set, DAYnnn processing will fail and can only be restarted from the beginning.

Note: You can override the RESTARTWORK data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the RESTARTWORK keyword on the continuation line.

RESTARTWORK accepts the engine/host options documented in "SAS Companion for the z/OS Environment", including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
  RESTARTWORK  STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

```
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated, where:

             xxxx is TRK, CYL, or blklen
             pp is the primary allocation
             ss is the secondary allocation

             and ROUND specifies that the allocated space be
             "rounded" to a cylinder boundary when the unit
             specified was a block length.  ROUND is ignored
             with the TRK or CYL options.
```

Example 2 (multiple lines):

```
  RESTARTWORK  STORCLAS=MICSTEMP UNIT=SYSDA
  RESTARTWORK  SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

```
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  UNIT     - specifies the generic unit for a new data set.
             The name can have up to 8 characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.
```

```
INCRUPDATE
----------

This statement is optional.  Specify this to enable
incremental update for this product:

INCRUPDATE YES

If you do not specify or enable the INCRUPDATE parameter,
then this option defaults to this and incremental update is
disabled:

INCRUPDATE NO

************************************************************
*                                                          *
*  Note:  Changing the INCRUPDATE parameter (either from NO *
*         to YES or from YES to NO) requires regeneration   *
*         of the DAILY operational job by executing         *
*         prefix.MICS.CNTL(JCLGEND) or by specifying        *
*         DAILY in prefix.MICS.PARMS(JCLGENU) and           *
*         executing prefix.MICS.CNTL(JCLGENU).              *
*                                                           *
*         If you specify INCRUPDATE YES, you must also      *
*         generate the INCRccc, cccIUALC, and cccIUGDG jobs *
*         (where ccc is the 3 character product ID).        *
*         Depending on the options you select, you may also *
*         need to execute the cccIUALC and/or cccIUGDG      *
*         jobs.                                             *
*                                                           *
************************************************************

Incremental update can significantly reduce time and resource
usage in the DAILY job by letting you split out a major
portion of daily database update processing into multiple,
smaller, incremental updates executed throughout the day.

o  Standard CA MICS database update processing involves (1)
   reading and processing raw input data to generate DETAIL
   and DAYS level CA MICS database files, followed by (2)
   summarization of DETAIL/DAYS level data to update
   week-to-date and month-to-date database files.

o  When you activate incremental update:

   -  You can execute the first-stage processing (raw data
      input to create DETAIL/DAYS files) multiple times
      throughout the day, each time processing a subset of
      the total day's input data.

   -  Then, during the final update of the day (in the
      DAILY job), the incremental DETAIL/DAYS files are
      "rolled-up" to the database DETAIL and DAYS
      timespans, and then summarized to update the
      week-to-date and month-to-date files.

o  Incremental update is independent of your internal step
   restart or DBSPLIT specifications.  You have the option
   to perform incremental updates with or without internal
   step restart support.

o  Incremental update is activated and operates
   independently by product.  The incremental update job
   for this product, INCRccc (where ccc is the product ID),
   can execute concurrently with the incremental update job
   for another product in the same unit database.

o  The CA MICS database remains available for reporting and
   analysis during INCRccc job execution.
```

```
*************************************************************
*                                                           *
*  Note:  CA MICS is a highly configurable system           *
*         supporting up to 36 unit databases, each of which *
*         can be configured and updated independently.      *
*         Incremental update is just one of the options you *
*         can use to configure your CA MICS complex.        *
*                                                           *
*         All efforts should be made to employ CA MICS      *
*         configuration capabilities to minimize issues     *
*         prior to activating incremental update.  For      *
*         example:                                          *
*                                                           *
*         o  Splitting work to multiple units is an         *
*            effective way to enable parallel database      *
*            update processing                              *
*                                                           *
*         o  Adjusting account code definitions to ensure   *
*            adequate data granularity while minimizing     *
*            total database space and processing time       *
*                                                           *
*         o  Tailoring the database to drop measurements    *
*            and metrics of lesser value to your            *
*            data center, thereby reducing database update  *
*            processing and resource consumption            *
*                                                           *
*         While incremental update is intended to reduce    *
*         DAILY job elapsed time, total resource usage of   *
*         the combined INCRccc and DAILY jobs steps can     *
*         increase due to the additional processing         *
*         required to maintain the incremental update       *
*         "to-date" files and for roll-up to the unit       *
*         database.  The increased total resource usage     *
*         will be more noticeable with small data volumes,  *
*         where processing code compile time is a greater   *
*         percentage of total processing cost.              *
*                                                           *
*************************************************************
```

Note: When you activate incremental update (INCRUPDATE YES), the following optional incremental update parameters are enabled. These parameters have no effect if incremental update is disabled (INCRUPDATE NO). For more details, see the individual parameter descriptions later in this section.

o INCRDB PERM/TAPE/DYNAM

o INCRDETAIL data_set_allocation_parameters

o INCRDAYS data_set_allocation_parameters

o INCRCKPT data_set_allocation_parameters

o INCRSPLIT USE/IGNORE data_set_allocation_parameters

Incremental update processing reads and processes raw measurement data to create and maintain DETAIL and DAYS level "to-date" files for the current day.

o These incremental update database files are maintained on unique z/OS data sets, independent of the standard CA MICS database files, and independent of any other product's incremental update database files. There is one data set each for DETAIL and DAYS level "to-date" data and a single incremental update checkpoint data set for this product in this unit.

o The incremental update DETAIL and DAYS files can be permanent DASD data sets, or they can be allocated dynamically as needed and deleted after DAILY job processing completes. Optionally, you can keep the incremental update DETAIL and DAYS files on tape, with the data being loaded onto temporary DASD space as needed for incremental update or DAILY job processing. See the INCRDB PERM/TAPE/DYNAM option for more information.

After activating incremental update, you will use three incremental update facility jobs found in prefix.MICS.CNTL (Note that ccc is the product ID):

o cccIUALC

You execute this job to allocate and initialize the incremental update checkpoint file, and optionally the incremental update DETAIL and DAYS database files. cccIUALC is generally executed just ONE time.

o cccIUGDG

You execute this job to add generation data group (GDG) index definitions to your system catalog in support of the INCRDB TAPE option. cccIUGDG is generally executed just ONE time.

o INCRccc

This is the job you execute for each incremental update. You will integrate this job into your database update procedures for execution one or more times per day to process portions of the total day's measurement data.

Note: The DAILY job is run once at the end of the day. It will perform the final incremental update for the day's

data, and then roll-up the incremental DETAIL/DAYS files to the database DETAIL and DAYS timespans and update the week-to-date and month-to-date files.

INCRUPDATE Considerations

o Overhead

Incremental update is intended to reduce DAILY job resource consumption and elapsed time by offloading a major portion of database update processing to one or more executions of the INCRccc job. In meeting this objective, incremental update adds processing in the INCRccc and DAILY jobs to accumulate data from each incremental update execution into the composite "to-date" DETAIL and DAYS incremental update files, and also adds processing in the DAILY job to copy the incremental update files to the unit database DETAIL and DAYS timespans. The amount of this overhead and the savings in the DAILY job are site-dependent, and will vary based on input data volume and on the number of times INCRccc is executed each day.

In addition, activating incremental update will cause additional compile-based CPU time to be consumed in the DAYnnn DAILY job step. The increase in compile time is due to additional code included for each file structure in support of the feature. This increase should be static based on the scope of the CA MICS data integration product in terms of files. This compile-time increase does not imply an increase in elapsed or execution time. Incremental update allows I/O bound, intensive processing (raw data inputting, initial CA MICS transformation, etc.) to be distributed outside of the DAILY job. I/O processing is the largest contributor to elapsed time in large volume applications. Thus, the expected overall impact is a decrease in the actual runtime of the DAYnnn job step.

o Increased "Prime Time" Workload

By offloading work from the DAILY job to one or more INCRccc executions throughout the day, you are potentially moving system workload and DASD work space usage from the "off-hours," (when the DAILY job is normally executed) to periods of the day where your system resources are in highest demand. You should schedule INCRccc executions carefully to avoid adverse impact to batch or online workloads. For example, if your site's "prime shift" is 8:00 AM to 5:00 PM, you might choose to schedule incremental updates for 7:00 AM (just before "prime shift") and 6:00 PM (just after "prime shift"), with the DAILY job executing just after midnight.

o Increased DASD Usage

The DASD space required for the incremental update DETAIL and DAYS database files is in addition to the DASD space already reserved for the CA MICS database. By default, the incremental update database files are permanently allocated, making this DASD space unavailable for other applications. In general, you can assume that the incremental update database files will require space equivalent to two cycles of this product's DETAIL and DAYS timespan files.

Alternatively, the incremental update database files can be allocated in the first incremental update of the day and deleted by the DAILY job (see the INCRDB DYNAM option later in this section). This approach reduces the amount of time that the DASD space is dedicated to incremental update, and lets the amount of DASD space consumed increase through the day as you execute each incremental update.

A third option is to store the incremental update database files on tape (see the INCRDB TAPE option). With this approach, the DASD space is required just for the time that each incremental update or DAILY job step is executing. Note that while this alternative reduces the "permanent" DASD space requirement, the total amount of DASD space required while the incremental update or DAILY jobs are executing is unchanged. In addition, the TAPE option adds processing to copy the incremental update files to tape, and to reload the files from tape to disk.

Note: The incremental update checkpoint file is always a permanently allocated disk data set. This is a small data set and should not be an issue.

o Operational Complexity

Incremental update expands your measurement data management and job scheduling issues. You must ensure that each incremental update and the DAILY job processes your measurement data chronologically; that is, each job must see data that is newer than the data processed by the prior job. By incrementally updating the database, you have more opportunities to miss a log file, or to process a log out of order.

o Interval End Effects

Each incremental update processes a subset of the day's measurement data, taking advantage of early availability of some of the day's data, for example, when a measurement log fills and switches to a new volume. This can cause a problem if the measurement log split occurs while the data source is logging records for the end of a measurement interval, thus splitting the data for a single measurement interval across two log files. When an incremental update processes the first log file, the checkpoint high end timestamp is set to indicate that this split measurement interval has been processed. Then, when the rest of the measurement interval's data is encountered in a later update, it can be dropped as duplicate data (because data for this measurement interval end timestamp has already been processed).

Appropriate scheduling of log dumps and incremental updates can avoid this problem. For example, if you plan to run incremental updates at 7:00 AM and 6:00 PM, you could force a log dump in the middle of the measurement interval just prior to the scheduled incremental update executions. This is an extension of the procedure you may already be using for end-of-day measurement log processing. The objective is to ensure that all records for each monitor interval are processed in the same incremental update.

o Dynamic Allocation

When you activate incremental update and specify TAPE or DYNAM for the INCRDB parameter, dynamic allocation is employed for the incremental update database files. If your site restricts dynamic allocation of large, cataloged data sets, you must use the INCRDETAIL and INCRDAYS parameters to direct incremental update data set allocation to a generic unit or storage class where dynamic allocation is allowed.

o Data Set Names

The incremental update database files are allocated and cataloged according to standard CA MICS unit database data set name conventions. The DDNAME and default data set names are (where ccc is the product ID):

o Incremental update checkpoint file,
```
//IUCKPT   DD DSN=prefix.MICS.ccc.IUCKPT,.....
```

o Incremental update DETAIL
```
//IUDETAIL DD DSN=prefix.MICS.ccc.IUDETAIL,.....
```

o Incremental update DAYS
```
//IUDAYS   DD DSN=prefix.MICS.ccc.IUDAYS,....
```

Since these data sets conform to the same data set name conventions as your existing CA MICS data sets, there should be few, if any, data-set-name-related allocation issues. However, it is possible to override the data set names if required. Contact Technical Support at http://ca.com/support for assistance if you must change data set names.

INCRDB

This statement is optional. The default is this:

INCRDB PERM

Note: INCRDB is ignored when you specify INCRUPDATE NO.

Specify this statement or take the default, to keep the incremental update database DETAIL and DAYS files on permanently allocated DASD data sets:

INCRDB PERM

Execute the prefix.MICS.CNTL(cccIUALC) job to allocate the incremental update database files.

```
************************************************************
*                                                          *
*  Note:  The incremental update checkpoint file is always *
*         a permanently allocated DASD data set.           *
*                                                          *
************************************************************
```

Specify this to offload the incremental update DETAIL and DAYS files to tape between incremental update executions:

INCRDB TAPE #gdgs UNIT=name

With the TAPE option, the incremental update DETAIL and DAYS DASD data sets are dynamically allocated at the beginning of the incremental update job or DAILY job step, and then are deleted after the job step completes.

o The first incremental update job of the day allocates and initializes the incremental update database files. At the end of the job, the DETAIL and DAYS files are

copied to a new (+1) generation of the incremental update tape data sets. Then the DASD files are deleted.

- o Subsequent incremental update jobs restore the DASD incremental update database files from the current, (0) generation, incremental update tape data sets before processing the input measurement data. At the end of the job, the DETAIL and DAYS files are copied to a new (+1) generation of the incremental update tape data sets. Then the DASD files are deleted.

- o The DAILY job step also restores the DASD incremental update database files from the (0) generation tape files before processing the input data, but does NOT copy the incremental update database files to tape. Thus, the DAILY job actually creates a new, null (+1) generation.

- o Use the #gdgs parameter to specify the maximum number of incremental update tape generations. The minimum is 2 and the maximum is 99, with a default of 5. Set the number of generations equal to or greater than the number of incremental updates, including the DAILY job you plan to execute each day. This facilitates restart and recovery if you encounter problems requiring you to reprocess portions of the daily measurement data.

- o Use the optional UNIT=name parameter to specify a tape unit name for the incremental update database output tapes. The default is to use the same tape unit as the input tapes.

- o A special index must be created in your system catalog for each of the incremental update tape data set generation data groups. The prefix.MICS.CNTL(cccIUGDG) job will generate the statements to create the incremental update GDG index definitions.

  - Before each index is built, it is deleted. These DLTX (or DELETE) statements causes an error message if no entry exists. This is done so that you can change the number of entries without having to delete each of the index entries.

  - DLTX and BLDG (or DELETE and DEFINE) fail if there is a cataloged data set with the same index. IDCAMS issues a message and gives a return code of 8. This issue is not a problem for non-GDG entries or if the GDG already has the desired number of entries.

  - If you want to change the number of entries kept in a GDG with cataloged data sets, do the following:

    1. Uncatalog any existing entries in the GDG.
    2. Delete the index with a DLTX (or DELETE).
    3. Create the index with a BLDG (or DEFINE).
    4. Catalog any entries that are uncataloged in step 1.

- o The incremental update tape data set names are as follows, where ccc is the product ID:

  - Incremental update tape DETAIL file
    tapeprefix.MICS.ccc.IUXTAPE.GnnnnV00

  - Incremental update tape DAYS file
    tapeprefix.MICS.ccc.IUDTAPE.GnnnnV00

```
************************************************************
*                                                          *
*  Note:  The INCRDETAIL and INCRDAYS parameters are       *
*         required when you specify INCRDB TAPE.           *
*                                                          *
************************************************************
```

Specify this parameter to allocate dynamically the incremental update DETAIL and DAYS DASD data sets in the first incremental update of the day, and then delete these data sets at the end of the DAILY job step:

INCRDB DYNAM

- o With this option, no space is used for the incremental update database files during the time between the end of the DAILY job step and the beginning of the next day's first incremental update.

- o With this approach, you can set the data set allocation parameters so that the incremental update DETAIL and DAYS data sets start out with a minimum allocation and then grow through secondary allocations as more space is required for subsequent incremental updates. For example, enough space for one incremental update.

```
************************************************************
*                                                          *
*  Note:  The INCRDETAIL and INCRDAYS parameters are       *
*         required when you specify INCRDB DYNAM.          *
*                                                          *
************************************************************
```

INCRDETAIL

This statement is required if you specify either of these:

INCRDB TAPE

INCRDB DYNAM

Otherwise, this statement is optional. There is no default.

Specify this to define data set allocation parameters for the incremental update DETAIL data set (IUDETAIL):

INCRDETAIL data_set_allocation_parameters

Note: INCRDETAIL is ignored when you specify INCRUPDATE NO.

The incremental update DETAIL data set (IUDETAIL) contains the current incremental update detail-level database files, and the DETAIL "to-date" data for the current daily update cycle. You should allocate DASD space equivalent to two cycles of this product's DETAIL timespan data.

If you specified INCRDB PERM (the default), your INCRDETAIL parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update database and checkpoint files.
- o Omit the INCRDETAIL parameter if you prefer to specify data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

If you specified INCRDB TAPE or INCRDB DYNAM, your INCRDETAIL parameter specifications are used in incremental update DETAIL data set dynamic allocation during incremental update or DAILY job step execution.

- o The INCRDETAIL parameter is required for the TAPE or DYNAM option.
- o Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRDETAIL keyword on the continuation line.
- o INCRDETAIL accepts the engine/host options documented in the SAS Companion for the z/OS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

  Important! DO NOT SPECIFY THE DISP PARAMETER.
- o You can override the INCRDETAIL data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Example 1:

```
INCRDETAIL   STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

```
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated, where:

             xxxx is TRK, CYL, or blklen
             pp   is the primary allocation
             ss   is the secondary allocation

             and ROUND specifies that the allocated space be
             "rounded" to a cylinder boundary when the unit
             specified was a block length.  ROUND is ignored
             with the TRK or CYL options.
```

Example 2 (multiple lines):

```
INCRDETAIL   STORCLAS=MICSTEMP UNIT=SYSDA
INCRDETAIL   SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

```
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  UNIT     - specifies the generic unit for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.
```

INCRDAYS

This statement is required if you specify either of these:

INCRDB TAPE

INCRDB DYNAM

Otherwise, this statement is optional. There is no default.

Specify this to define data set allocation parameters for the incremental update DAYS data set (IUDAYS):

```
INCRDAYS      data_set_allocation_parameters
```

Note: INCRDAYS is ignored when you specify INCRUPDATE NO.

The incremental update DAYS data set (IUDAYS) contains the current incremental update days-level database files, and the DAYS "to-date" data for the current daily update cycle. You should allocate DASD space equivalent to two cycles of this product's DAYS timespan data.

If you specified INCRDB PERM (the default), your INCRDAYS parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update database and checkpoint files.
- o Omit the INCRDAYS parameter if you prefer to specify data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

If you specified INCRDB TAPE or INCRDB DYNAM, your INCRDAYS parameter specifications are used in incremental update DAYS data set dynamic allocation during incremental update or DAILY job step execution.

- o The INCRDAYS parameter is required for the TAPE or DYNAM option.
- o Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRDAYS keyword on the continuation line.
- o INCRDAYS accepts the engine/host options documented in the SAS Companion for the z/OS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

  Important! DO NOT SPECIFY THE DISP PARAMETER.
- o You can override the INCRDAYS data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, Section 2.3.6.

Example 1:

```
INCRDAYS     STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

```
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated, where:

             xxxx is TRK, CYL, or blklen
             pp   is the primary allocation
             ss   is the secondary allocation

             and ROUND specifies that the allocated space be
             "rounded" to a cylinder boundary when the unit
             specified was a block length.  ROUND is ignored
             with the TRK or CYL options.
```

Example 2 (multiple lines):

```
INCRDAYS     STORCLAS=MICSTEMP UNIT=SYSDA
INCRDAYS     SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

```
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  UNIT     - specifies the generic unit for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.
```

INCRCKPT

This statement is optional. Specify this to override default data set allocation parameters for the incremental update checkpoint data set:

```
INCRCKPT      data_set_allocation_parameters
```

Note: INCRCKPT is ignored when you specify INCRUPDATE NO.

The incremental update checkpoint data set tracks incremental update job status and the data that has been processed during the current daily update cycle. The incremental update checkpoint is used to detect and block the input of duplicate data during incremental update processing. This data set will be exactly the same size as prefix.MICS.CHECKPT.DATA (the unit checkpoint data set), usually 20K to 200K depending on the prefix.MICS.PARMS(SITE) CKPTCNT parameter (100-9999).

Your INCRCKPT parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update checkpoint file. If you specified INCRDB PERM, then the cccIUALC job will also allocate the incremental update DETAIL and DAYS database files.
- o By default the incremental update checkpoint data set is allocated as SPACE=(TRK,(5,2)) using the value you specified for the prefix.MICS.PARMS(JCLDEF) DASDUNIT parameter.
- o Omit the INCRCKPT parameter if you prefer to override data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRCKPT keyword on the continuation line.

INCRCKPT accepts the engine/host options documented in the SAS Companion for the MVS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
INCRCKPT     STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

```
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated, where:

             xxxx is TRK, CYL, or blklen
             pp   is the primary allocation
             ss   is the secondary allocation

             and ROUND specifies that the allocated space be
             "rounded" to a cylinder boundary when the unit
             specified was a block length.  ROUND is ignored
             with the TRK or CYL options.
```

Example 2 (multiple lines):

```
INCRCKPT     STORCLAS=MICSTEMP UNIT=SYSDA
INCRCKPT     SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

```
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  UNIT     - specifies the generic unit for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.
```

INCRSPLIT

This statement is optional and defaults to this:

INCRSPLIT IGNORE

Specify the following if you want the incremental update job for this product to get input measurement data from the output of the SPLITSMF job. The optional data_set_allocation_parameters are used by the SPLITSMF job when creating the measurement data file for this product.

INCRSPLIT USE data_set_allocation_parameters

Note: INCRSPLIT is ignored when you specify INCRUPDATE NO.

This option would be used when multiple products in a single unit database are enabled to incremental update. The SPLITSMF job performs the same function for incremental

update jobs as the DAILY job DAYSMF step performs for the DAYnnn database update steps.

- o The SPLITSMF job dynamically allocates, catalogs, and populates prefix.MICS.ccc.IUSPLTDS data sets for each product in the unit database for which you specified both the INCRUPDATE YES and INCRSPLIT USE parameters. These data sets are then deleted after processing by the appropriate INCRccc job.

- o Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRSPLIT keyword on each continuation line.

- o INCRSPLIT accepts the engine/host options documented in the SAS Companion for the MVS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

  Important! DO NOT SPECIFY THE DISP PARAMETER.

Specify the following or accept the default if you want the incremental update jobs for this product to get their input measurement data from the data sets specified in the INPUTccc (or INPUTSMF) member of prefix.MICS.PARMS:

```
INCRSPLIT  IGNORE
```

When you specify INCRSPLIT IGNORE, this product will NOT participate in SPLITSMF job processing.

Example 1:

```
INCRSPLIT USE  STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen
pp is the primary allocation
ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRSPLIT  USE  STORCLAS=MICSTEMP UNIT=SYSDA
INCRSPLIT       SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set. The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated.

SPECIAL CONSIDERATIONS:

There are a number of files in the VAX/VMS Analyzer database that contain metrics which are the result of "delta" calculations done from one interval to the next. In these files, the first interval encountered on any given processing run is used to initialize the values for the subsequent "delta" calculations. Because of this, that first time interval in most cases will NOT be represented in the output file.

This architecture is acceptable as it only comes into play once for each DAILY processing cycle. When the Incremental Update feature is activated, this condition may become more noticeable, depending on the number of increments in any given DAILY processing cycle.

It is important to schedule your incremental cutoffs at off peak times to avoid the reporting impacts of a missing interval.

The following VAX/VMS Analyzer files have been identified as being susceptible to the condition described above:

DEMDSK - VMS MONITOR Disk File
DEMMOD - VMS MONITOR Modes File
DEMPRO - VMS MONITOR Process File
DEMSCS - VMS MONITOR SCS File

```
     DEMSPR - VMS System Profile File
     DESPRX - VMS Process Activity File

Another important area of consideration is the processing of
DEXSUS data.  DEXSUS data is created by running a script of
commands on VAX machine once a day which takes a snapshot of
the system environment.  Since this is a once a day activity,
the DEXSUS data will only be processed in the DAILY
processing step DAY074.  The DEXSUS data will not be
processed in the INCRDEX job even though the DD for the data
is present.

DYNAMWAIT
---------

This statement is optional.  Specify the following:

      DYNAMWAIT   minutes

to override the default amount of time, in minutes, the DAILY
and/or INCRccc job will wait for an unavailable data set.

Note:  This optional parameter is not normally specified.
       The system default is adequate for most data centers.

Internal Step Restart and Incremental Update facilities use
z/OS dynamic allocation services to create new data sets and
to access existing data sets.  Data set naming conventions
and internal program structure are designed to minimize data
set contention.  However, if data set allocation does fail
because another batch job or online user is already using a
data set, DAILY and/or INCRccc processing will wait 15
seconds and then try the allocation again.  By default, the
allocation will be attempted every 15 seconds for up to 15
minutes.  After 15 minutes, the DAILY or INCRccc job will
abort.

If data set contention in your data center does cause
frequent DAILY or INCRccc job failures, and you are unable to
resolve the contention through scheduling changes, you may
want to use the DYNAMWAIT parameter to increase the maximum
number of minutes the DAILY and/or INCRccc jobs will wait for
the data set to become available.

On the other hand, if your data center standards require
that the DAILY and/or INCRccc jobs fail immediately if
required data sets are unavailable, specify the following:

      DYNAMWAIT 0

Note:  You can override the DYNAMWAIT parameter at
       execution-time using the //PARMOVRD facility.  For
more information about execution-time override of
dynamic data set allocation parameters, see the PIOM,
section 2.3.6.
```

```
                          +-------------------------------------------------------------------------+
                          | INSTALLATION PREPARATION WORKSHEET: VAX/VMS Unit Options Definitions     |
                          |                                                                         |
                          | PARMS Library Member is DEXOPS                                          |
                          | Reference Sections: 7.3.1, 9.4                                          |
                          +-------------------------------------------------------------------------+
                          |                                                                         |
                          | DEX PROCESSING OPTIONS:                                                 |
                          |                                                                         |
                          |                                                                         |
                          |                                                                         |
  System-oriented options |    OPTIONS  _______(orgsysid)  ________(nodename)                       |
                          |                                                                         |
                          |    MONITOR  _______(orgsysid)  ________(ddname)                         |
                          |                                                                         |
                          |    ACCOUNT  _______(orgsysid)  ________(ddname)                         |
                          |                                                        (default         |
                          |    DEXSUS   _______(orgsysid)  ________(ddname)  ________  duration)     |
                          |                                                                         |
                          |                                                                         |
                          |                                                                         |
  General-unit option     |    VERIFY     _______(ABORT/NOABORT)                                    |
                          |                                                                         |
  Disk Accounting option  |    DISKACCT   _______(DETAIL/DAYS)                                      |
                          |                                                                         |
                          |                                                                         |
                          |                                                                         |
  Special-purpose option  |    COMMON     _______(datatype)         ________(ddname)  _____(EXIT)   |
                          |                      (MONITOR/                                          |
                          |                       ACCOUNT/                                          |
                          |                       DEXSUS)                                           |
                          |                                                                         |
                          |    WORK       _______ n data_set_allocation_parmeters                   |
                          |                                                                         |
                          |    MULTWORK|NOMULT __ fff ... fff                                       |
                          |                                                                         |
                          |    RESTART    _______ YES/NO                                            |
                          +-------------------------------------------------------------------------+
                          | ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
                          +-------------------------------------------------------------------------+
```

Figure 7-3. VAX/VMS Analyzer Options Definition Worksheet

7.3.2 - Input DD Statements for VAX/VMS (INPUTDEX)

The CA MICS VAX/VMS Analyzer processes three different types of input: VMS MONITOR data, VMS ACCOUNTING data, and CA MICS DEXSUS data. These files must be transported to the MVS system that runs CA MICS. You must provide the process for moving the data from your VMS system to your MVS system.

The input DD statements to define the VAX/VMS input files must be supplied in the INPUTDEX member of the prefix.MICS.PARMS library and are referenced by the MONITOR, ACCOUNT, and DEXSUS statements of DEXOPS. A worksheet for coding this member is shown in Figure 7-4.

It is quite likely that you will have data from multiple systems as input to the update process. To provide for data from multiple systems, provide one DD statement per system, using the DDname(s) specified in DEXOPS.

There must be one DD statement in the INPUTDEX member for each VAX/VMS data source identified in the prefix.MICS.PARMS(DEXOPS) library member. The order in which the DDnames are specified is not important. The order of data sets WITHIN a DDname, however, must be chronological. Do not use a GDG base because the system will order the generations in the GDG base backwards with the most current generation being input first. This condition will cause the VAX/VMS logs to be processed out of chronological order, which is a condition we wish to avoid.

For example, if your VAX/VMS data were to be read from an NL tape volume VAX001 and input through DDnames DEXMON01, DEXACC01, and DEXSUS01 (DDnames are defined in MICS.PARMS(DEXOPS)), you would change the INPUTDEX member to contain the following:

```
//DEXMON01 DD DISP=OLD,VOL=SER=VAX001,LABEL=(1,NL),
//         DCB=(RECFM=VB,LRECL=32752,BLKSIZE=32756)
//DEXACC01 DD DISP=OLD,VOL=SER=VAX001,LABEL=(2,NL),
//         DCB=(RECFM=VB,LRECL=32752,BLKSIZE=32756)
//DEXACC01 DD DISP=OLD,VOL=SER=VAX001,LABEL=(3,NL),
//         DCB=(RECFM=VB,LRECL=32752,BLKSIZE=32756)
```

For another example, in the JCL below, the unit will process VMS ACCOUNTING data from SYS1 and SYS2, VMS MONITOR data from SYS1, SYS2, and SYS3, and DEXSUS data from SYS1. The DD statements will read as follows:

```
//DEXACC01 DD DISP=SHR,DSN=VMS.SYS1.ACCOUNT.DATA
//DEXACC02 DD DISP=SHR,DSN=VMS.SYS2.ACCOUNT.DATA
//DEXMON01 DD DISP=SHR,DSN=VMS.SYS1.MONITOR.DATA
//DEXMON02 DD DISP=SHR,DSN=VMS.SYS2.MONITOR.DATA
//DEXMON03 DD DISP=SHR,DSN=VMS.SYS3.MONITOR.DATA
//DEXSUS01 DD DISP=SHR,DSN=VMS.SYS1.DEXSUS.DATA
```

NOTE: Because of the text substitution process used by JCLGEN, to produce '//*', '/*', and '&' in the JCL produced by JCLGEN, the following symbols must be used instead:

//@	produces	//*
/@	produces	/*
?	produces	&

DCB Parameters

DCB record description parameters are required only when data is provided to the DAY074 step by unlabeled tape.

VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS data all have a RECFM=VB. You can use the blocksize used by the creating program, but this is often difficult to determine and may vary from day to day. For all three types of data files, the safest, easiest, and least error-prone approach is to use RECFM=VB,LRECL=32752,BLKSIZE=32756.

Providing Data to CA MICS

The CA MICS system administrator is responsible for getting the data to be processed each day into the VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS data sets. The preferred methods are to use some form of file transmission such as CA FileExchange software, the DECnet/SNA Data Transfer Facility (DTF), or some similar facility that allows you to transfer files between your VAX/VMS systems and your MVS systems. If no other method is available, you can move files by using tape. To learn more about the record types processed by CA MICS and how to transport the VMS data files to MVS, refer to Chapter 6, Data Sources, in this guide.

```
INSTALLATION PREPARATION WORKSHEET: INPUTDEX JCL Definitions

PARMS Library Member is INPUTDEX
Reference Section:  7.3.2


This definition is required to specify the DD statements for VAX/VMS
data that will be read by the DAILY CA MICS job.

  //@
  //@ WARNING: ALWAYS MAKE CHANGES IN PARMS(INPUTDEX) AND NOT
  //@          &CNTL(DAILY).
  //@          CHANGES MADE TO &CNTL(DAILY)
  //@          WILL BE GONE WHEN DAILY REGENERATED BY JCLGEN.
  //@
  //@ SPECIFY THE INPUT MONITOR AND ACCOUNTING DATA SETS FOR ALL
  //@   SYSTEMS
  //@
  //@ ACCOUNTING DATA
  //DEXACC01 DD DISP=SHR,DCB=BUFNO=3,DSN=______________________
  //DEXACC02 DD DISP=SHR,DCB=BUFNO=3,DSN=______________________
  //DEXACC03 DD DISP=SHR,DCB=BUFNO=3,DSN=______________________
  //@ MONITOR DATA
  //DEXMON01 DD DISP=SHR,DCB=BUFNO=3,DSN=______________________
  //DEXMON02 DD DISP=SHR,DCB=BUFNO=3,DSN=______________________
  //DEXMON03 DD DISP=SHR,DCB=BUFNO=3,DSN=______________________
  //@ DEXSUS  DATA
  //DEXSUS   DD DISP=SHR,DCB=BUFNO=3,DSN=______________________
```

Figure 7-4. INPUTDEX JCL Definitions Worksheet

7.3.3 - Input Needed to Generate VAX System Code (DEXPGEN)

Some CA MICS code for the CA MICS VAX/VMS Analyzer is generated at the unit level through the DEXPGEN job. The applicable input prefix.MICS.PARMS members that must be completed prior to DEXPGEN execution are identified in the parentheses below:

- o VAX/VMS processing options definition (DEXOPS)
- o Input DD statements for VAX/VMS (INPUTDEX)

Figure 7-5 identifies for the VAX/VMS unit level code generation process the prefix.MICS.PARMS input members, output members, and SAS macro names (a member may contain the definition of more than one SAS macro). The output libraries are at the data base unit level (prefix).

prefix.MICS.PARMS Input Member	USER.SOURCE Output Members and SAS Macro Names	USER.LOAD Output Member
DEXOPS,	$DEXMSTR	$DEXSYS
INPUTDEX,	%DEXINCL	$DEXDUR
SYSID	%DEXINFA	
	%DEXINFM	
	%DEXINFS	
	%DEXINPTA	
	%DEXINPTM	
	%DEXINPTS	
	%DEXINPA	
	%DEXINPM	
	%DEXINPS	
	%DEXCHKA	
	%DEXCHKM	
	%DEXCHKS	

Figure 7-5. Members/Macros Needed/Updated by DEXPGEN Code

7.3.4 - Database Space Modeling (DBMODEL)

This section describes the information required to define the Analyzer to the Database Space Modeling Facility. Specifically, the user must provide values for the cycle (data retention) definitions. The special considerations for defining the FILE input statements to the Database Space Modeling facility are discussed.

Section 2.3.4 of the PIOM, Database Space Modeling Facility, explains how these values are used in estimating the DASD requirements of the database, and how the user can input these values to the modeling facility.

This section covers the following topics:

1 - Data Retention Specifications (FILE Statements)
2 - DBMODEL Input Statements

7.3.4.1 - Data Retention Specifications (FILE Statements)

Figure 7-6 provides a worksheet to define the different retention limits for the online and archive data base files. The numbers shown on the worksheet reflect the default retention values in prefix.MICS.PARMS(DBMODEL).

Use the worksheet to define your retention specifications, which, in turn, will allow you to determine the appropriate values for the modeling process.

Note that the DBMODEL worksheet shown here contains values for this product only. See Section 2.3.4.1, Preparing the Modeling Input, in the PIOM for additional information.

The worksheet is organized by information area. Each file in the area is listed by name. For each file, a line is formatted to allow six definitions in the online database and two in the archive database:

- o The online database files quantify the number of cycles of data that will be maintained in the DETAIL, DAYS, WEEKS, MONTHS, and YEARS timespans and the TABLES data area.

- o The two definitions for the archive database files quantify the number of cycles of data to be retained, up to the cutoff limit defined. The archive definitions have no impact on the size of the database and may be specified whether or not the weekly and/or monthly archive history files have actually been activated (see Section 2.3.3, CA MICS JCL Planning and Parameters, of the PIOM).

The worksheet formats provide an underscored area for the user's definition, followed by the recommended value, shown within parentheses. If the underscored area contains a value of 00, the file is not supported for the indicated timespan. To add support, you must perform database tailoring as described in Section 6.2, Tailoring the Database, of the System Modification Guide (SMG).

When specifying a retention limit, remember that the number may never be zero if the file has been defined to be active in the timespan.

```
+-------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET: Database Data Retention Definitions  |
|                                                                          |
| PARMS Library Member is DBMODEL                                          |
| Reference Sections:  7.3.4.2                                             |
+------+-----------------------------------------------+-------------------+
|      |           Online Database Retention           |  Archive Cut-Off  |
| File |DETAIL    DAYS   WEEKS   MONTHS  YEARS  TABLES | WEEKS    MONTHS   |
| Name | __(NA)  __(NA)  __(NA)  __(NA)  _(NA)  __(NA) |___ (NA)  ___ (NA) |
+------+-----------------------------------------------+-------------------+
|DEAPRC| __(10)  __(00)  __(00)  __(13)  _(05)  00(00) |___ (000) ___ (036)|
|DEAIMG| __(10)  __(00)  __(00)  __(06)  _(00)  00(00) |___ (000) ___ (036)|
|DEAINT| __(10)  __(00)  __(00)  __(00)  _(00)  00(00) |___ (000) ___ (000)|
|DEA_LF| __(10)  __(00)  __(00)  __(13)  _(05)  00(00) |___ (000) ___ (036)|
|DEAPRQ| __(10)  __(00)  __(00)  __(13)  _(05)  00(00) |___ (000) ___ (036)|
+------+-----------------------------------------------+-------------------+
|DEMPRO| __(01)  __(07)  __(13)  __(13)  _(00)  00(00) |___ (053) ___ (036)|
|DEMMOD| __(01)  __(07)  __(13)  __(13)  _(00)  00(00) |___ (053) ___ (036)|
|DEMDSK| __(01)  __(07)  __(13)  __(13)  _(00)  00(00) |___ (053) ___ (036)|
|DEMSCS| __(01)  __(07)  __(13)  __(13)  _(00)  00(00) |___ (053) ___ (036)|
|DEMRMS| __(01)  __(07)  __(13)  __(13)  _(00)  00(00) |___ (053) ___ (036)|
|DEMSPR| __(01)  __(07)  __(13)  __(13)  _(00)  00(00) |___ (053) ___ (036)|
+------+-----------------------------------------------+-------------------+
|DESDKD| __(01)  __(07)  __(13)  __(13)  _(00)  00(00) |___ (053) ___ (036)|
|DESDKQ| __(01)  __(07)  __(13)  __(13)  _(00)  00(00) |___ (053) ___ (036)|
|DESDKU| __(02)  __(02)  __(00)  __(13)  _(05)  00(00) |___ (000) ___ (036)|
|DESPRX| __(01)  __(07)  __(13)  __(13)  _(03)  00(00) |___ (053) ___ (036)|
|DESSYU| __(01)  __(14)  __(53)  __(24)  _(00)  00(00) |___ (053) ___ (036)|
+------+-----------------------------------------------+-------------------+
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-------------------------------------------------------------------------+
```

Figure 7-6. Database Data Retention Definitions Worksheet

7.3.4.2 - DBMODEL Input Statements

The DBMODEL member of prefix.MICS.PARMS provides the input to the Database Space Modeling Facility.

Using the information collected on the worksheet in Figure 7-6, code the DBMODEL member. To actually perform the space modeling, submit the jobs as described in the CA MICS Planning, Installation, Operation, and Maintenance Guide, Section 2.3.4.2.

7.4 - Other Related Parameters

This section contains information on other parameters related to the operation and use of the CA MICS VAX/VMS Analyzer. CA MICS requires that Systems to be input be identified to ensure that your expectations in using the VAX/VMS Analyzer are met. Thus you must make sure that the parameter Computing System Parameters (SYSID) has an entry for each VAX/VMS system to be input.

If you use CA MICS Accounting and Chargeback, you should review the following definitions:

- o Charging Elements for VAX/VMS
- o Data collection method effects on charging
- o Rounding effects on charging

These areas are discussed further in the remainder of this section.

Computing System Parameters (SYSID)

Data in the CA MICS database are associated with and summarized according to the computing system that produced the raw data used to create the database data. Each computing system is identified by its SYSID.

Information about the SYSIDs used at your installation is coded in prefix.MICS.PARMS(SYSID). Changing this information after CA MICS is in use is a two-step process: first, change the SYSID member in prefix.MICS.PARMS and second, execute the job BASPGEN in the prefix.MICS.CNTL library.

CA MICS may either exclude data or terminate with a user abend if input data comes from a SYSID that is not defined in the PARMS library. The choice of exclusion or termination is discussed completely in Section 2.3.2.2 of the CA MICS Planning, Installation, Operation, and Maintenance Guide.

The following text summarizes how to code parameters for prefix.MICS.PARMS(SYSID):

- o The format of the statements is free-form but positional. Data element placement is as follows:
  - ORGSYSID (should match DEXOPS definitions)
  - Logical SYSID
  - CPU Model (enclosed in quotes)
  - SCP (VMS, VAXVMS, or *)
  - RMF CPU Conversion Factor (set to . or 0)
  - MIPS Rate (Any base unit of performance--VUPs, SpecMarks etc.)
    where VUP rating for VAX 11/780 = 1
  - System Name (1 to 30 character name enclosed in quotes)
- o A sample entry in SYSID:

```
VMS1 VMS1 '11/780' VMS 0 1.0 'VMS SYSTEM ONE'
```

o Considerations for the CA MICS VAX/VMS Analyzer:

- When coding SYSID parameter for a VMS system, Logical SYSID may be any value, although it is typically the same as the ORGSYSID. System entries in SYSID can be "autodefined". In this process the ORGSYSID values defined in DEXOPS will be automatically added to the SYSID Parameter (see the Section 2.3.2.2 of the CA MICS Planning, Installation, Operation and Maintenance Guide).

- VMS does not use the RMF Conversion Factor. It is necessary to code a value for this parameter, even for VMS systems, as a place holder in the SYSID statement.

- The MIPS rate is optional (see note below). It is necessary to have a value for this parameter as a place holder in the SYSID statement.

IMPORTANT NOTE
==========

The "MIPS Rate" may employ any base unit of performance and is used to populate an element known as fffCPUNI (where fff can refer to several files) representing the number of instructions executed. These elements simply represent CPU time multiplied by the "MIPS Rate" (base) factor specified. As a different rate can be employed in each system entry the resultant elements can be used as a comparative measure of work independent of machine speed. The value if specified may be derived from hardware vendors, comparative industry surveys, or benchmark studies conducted at your site. Site benchmarks will be the best source of input however they must be maintained over time and can be intensive in terms of processing and staffing. Typically any base factor(MIPS, VUPS(VMS units of Performance), SpecMarks etc.) will ultimately have been derived from a benchmark study on a specific set of work and workload mix effects the ratings.

Charging Elements for VAX/VMS

The CA MICS Accounting and Chargeback Guide describes the specific charging elements available for VAX/VMS resource usage. In general, these elements are supplied from the files in the VMS Accounting information area and from the Disk Usage (DKU) file in the VMS System Usage information area.

Data Collection Method Effects on Charging

The frequency and approach you choose for source data collection from VMS can affect the accuracy of certain charges for VMS usage. Our recommendations:

o Avoid collecting duplicate data from multiple systems for such resources as disk space on shared disks. Be sure you are not feeding the same data into CA MICS under multiple SYSIDs.

o Be aware of the impact of data collection frequency on disk occupancy statistics. Disk occupancy is calculated in megabyte hours, which has space and duration components. Duration is affected by frequency of data collection. In cases where duration is relatively short, occupancy measures are more susceptible to rounding to zero for the detail measurements.

Rounding Effects on Charging

For VMS disk usage charging, review your definition of the number of decimals for rounding in CA MICS Accounting and Chargeback. This value is set in the Accounting Options dialog, under the General Specifications option. A rounding value of 4 decimal places should be adequate to maintain accuracy for even small VMS files, depending on the charges per megabyte hour.

Be sure to review the DISKACCT option of DEXOPS. Your setting for this parameter may have an impact on the rounding effects you will experience. See section 7.3.1, VAX/VMS Processing Options (DEXOPS).

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 8

INSTALLATION

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 8

INSTALLATION

CHAPTER 8. INSTALLATION

After specifying the parameters documented in Chapter 7 of this guide, you can install the CA MICS VAX/VMS Analyzer using the checklists in Section 3.8 of the PIOM.

Be sure to review Chapters 6 and 7 of this guide before proceeding to the installation checklists in the CA MICS PIOM Guide. These chapters provide helpful information about the data sources and the input parameters you will need to supply for proper operation of the CA MICS VAX/VMS Analyzer.

In order to collect CA MICS VMS System Usage (DEXSUS) data you must install the DEXSUS.COM routine on your VMS system. This topic is covered in the following section:

1 - Installing the DEXSUS.COM Collection Routine

8.1 - Installing the DEXSUS.COM Collection Routine

The DEXSUS.COM collection routine is delivered in library sharedprefix.MICS.SOURCE(DEXSUS). To install this routine you must copy the DEXSUS member from your MVS system to a suitable directory on your VAX/VMS system. The file should be stored on VMS as DEXSUS.COM and is invoked by the command @DEXSUS from VMS.

A sample routine to invoke DEXSUS as a VMS batch job is provided with the VAX/VMS Analyzer as an example. This file is located in sharedprefix.MICS.SOURCE(DEXSUSJ1), and can be installed on VMS as DEXSUSJ1.COM.

The DEXSUS and DEXSUSJ1 routines are delivered as members of the MICS.SOURCE library on your MVS system in EBCDIC format. The operation to copy the DEXSUS and DEXSUSJ1 routines must translate the files from EBCDIC to ASCII.

The DEXSUSJ1 routine illustrates the use of several options for DEXSUS execution. These options are set externally from the DEXSUS routine, through VMS symbol assignments. Chapter 6 of this guide describes the operation of DEXSUS and gives examples of using DEXSUSJ1 to drive this routine as a batch job. Refer to section 6.1.1, "Data Collector Operations" for more explanation on this topic.

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 9

PROCESSING

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 9

PROCESSING

Chapter 9. PROCESSING

Creating and maintaining a large database system such as CA MICS requires a closely monitored approach to the periodic updating processes. The CA MICS database, just like any other information database (such as inventory control or personnel), requires a systematic, monitored update process.

It is vital to the system's effectiveness that the database be updated daily to build the required daily, week-to-date, month-to-date, and year-to-date files accurately. This data collection process for the CA MICS VAX/VMS Analyzer is vital to the proper operation of this product.

The processing flow of the CA MICS VAX/VMS Analyzer is described in the following sections:

1 - Processing Considerations
2 - Daily Update Processing Flow
3 - User Exits
4 - Using COMMON Input DDnames and DEXINPUT exits

9.1 - Processing Considerations

The DAY074 step of the Daily Update job is the step that processes VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS data. The input data from these three data sources is presented to the CA MICS VAX/VMS Analyzer in a series of sequential files.

VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS data files each consist of a series of variable length data records written on the VAX system. Before CA MICS can process the data, it must be transported to the MVS system on which CA MICS is running.

VMS ACCOUNTING data volume is significantly smaller than VMS MONITOR or CA MICS DEXSUS data. The CA MICS Accounting and Chargeback user cannot control the type of data generated and all data should be processed.

VMS MONITOR data can be quite voluminous. Processing this data can require large amounts of CPU time and DASD. You should carefully consider the types of data you wish to capture and process on the VMS system.

CA MICS DEXSUS data can be quite voluminous. Processing this data can require large amounts of CPU time and DASD. You should carefully consider the types of data you wish to capture and process on the VMS system.

U300 Abend

If any input process (VMS ACCOUNTING, VMS MONITOR, or CA MICS DEXSUS) encounters a situation in which no data has been read and kept when data input was specified for that source, a U300 abend will be produced. Refer to Section 4.3.11, Abend Codes and Corrective Action, of the CA MICS PIOM Guide to correct the abend.

9.2 - Daily Update Processing Flow

The major concepts and processes involved in collecting the input VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS data and updating the CA MICS database are described in the following sections:

1 - Input Raw Data
2 - Information Area Processing
3 - Database Timespan Processing
4 - File Aging

9.2.1 - Input Raw Data

In this phase, the raw VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS data is read, any nonprocessable (i.e., user-written) records are dropped; the input data fields are read, decoded, processed, and formatted to create the CA MICS data record; and the input data fields are written to the appropriate work file.

Figures 9-1a and 9-1b show the relationship between the VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS record types and the VAX/VMS Accounting, VAX/VMS Monitor, and VAX/VMS System Usage information area files:

VMS ACCOUNTING Records Name	Type	VAX/VMS Accounting Files	VAX/VMS System Usage Files
PROCESS TERMINATION	1	DEAPRC	DESPRX
IMAGE TERMINATION	3	DEAIMG	
SYSTEM INITIALIZATION	5	DEAINT	
LOGIN FAILED	7	DEA_LF	
PRINT QUEUED	8	DEAPRQ	

VMS MONITOR Records Name	Type	VAX/VMS Monitor Files	VAX/VMS System Usage Files
PROCESSES	0	DEMPRO	DESPRX
STATES	1	DEMSPR	
MODES	2	DEMMOD	
PAGE	3	DEMSPR	
IO	4	DEMSPR	
FCP	5	DEMSPR	
POOL	6	DEMSPR	
LOCK	7	DEMSPR	
DECNET	8	DEMSPR	
FILE SYSTEM CACHE	11	DEMSPR	
DISK	12	DEMDSK	
DLOCK	14	DEMSPR	
SCS	15	DEMSCS	
SYSTEM	17	DEMSPR	
CLUSTER	19	DEMSPR	
RMS	20	DEMRMS	
MSCP SERVER	21	DEMSPR	
TRANSACTION	22	DEMSPR	
VECTOR	23	DEMMOD	
SYSTEM IDENTIF.	129	--	DESSYU

Figure 9-1a. Record/File Correspondence

CA MICS DEXSUS Records

Name	Type		VAX/VMS System Usage Files
DEXSUS FILE HEADER	00		DESSYU
DEXSUS SYSTEM INFORMATION	01	ENDTS	all
VMS SHOW SYS	--	DEXNODE	all
DEXSUS SYI1	11		DESSYU
DEXSUS SYI2	12		DESSYU
DEXSUS SYI3	13		DESSYU
DEXSUS SYI4	14		DESSYU
DEXSUS SYSTEM MEMORY 1	20		--
VMS SHOW MEM/FULL/POOL	--		DESSYU
DEXSUS SYSTEM MEMORY 2	21		--
VMS SHOW MEM/PHY	--		DESSYU
DEXSUS DISK DEVICE	30	DEXDEVIC	DESDKD
VMS SHOW Device	--		DESDKD
DEXSUS DISK QUOTA	32	DEXDEVIC	DESDKQ
VMS Disk Quota	--	UIC records	DESDKQ
DEXSUS DISK USAGE	33	DEXDEVIC	DESDKU
VMS ANALYZE/DISK	--	01 records	DESDKD
VMS ANALYZE/DISK	--	02 records	DESDKU
DEXSUS FILE TRAILER	99		DESSYU

Figure 9-1b. Record/File Correspondence

USER EXITS

During this phase the USRSEL Exit and the USRSfff Exits are invoked. These exits are described further in section 9.3 "User Exits".

9.2.2 - Information Area Processing

The modules DYDEXFM1, DYDEXFM2, and DYDEXFM3 in sharedprefix.MICS.SOURCE drives the input for the VMS MONITOR, VMS ACCOUNTING, and CA MICS DEXSUS data. The actual input processing code is contained in the DEXACT module in sharedprefix.MICS.SOURCE for ACCOUNTING read-up, in the DEXMON module in sharedprefix.MICS.SOURCE for MONITOR read-up, and in the DEXSUSA module in sharedprefix.MICS.SOURCE for DEXSUS read-up. The history file (CKPTDATA) is created from separate files created in the read-up steps.

Physical structure: these modules are constructed at compile time by the macro %DEXINCL, which is generated into module $DEXMSTR in prefix.MICS.USER.SOURCE by DEXPGEN. %DEXINCL contains the %INCLUDE for DEXACT, DEXACTB (process ACCOUNTING data), DEXMON, DEXMONB (process MONITOR data), and DEXSUSA, DEXSUSB, DEXSUSC (process CA MICS DEXSUS data) when these data sources have been specified in prefix.MICS.PARMS member DEXOPS. When a data source has not been specified, a stub module (DEXACTST, DEXMONST, DEXSUSST) is included to create the 00 cycle files in DETAIL. %DEXINCL also performs a save function for the ADMIHL (CKPTDATA) file so that all input recording can be seen by CKRTN4.

The CA MICS VAX/VMS Analyzer differentiates between VMS versions' MONITOR records because record formats differ between versions.

The various recent versions of VMS ACCOUNTING have the same record format, with a minor change for VMS 6.1, indicated by a special flag bit.

The CA MICS VAX/VMS Analyzer differentiates between versions of CA MICS DEXSUS records, based on the VMS level and CA MICS DEXSUS maintenance level reported by DEXSUS.

User Exits

During this phase, the USRAfff Accounting Exits are invoked. These exits are described further in section 9.3 "User Exits."

9.2.3 - Database Timespan Processing

Using the newly created DETAIL or DAYS files, the DAYS, WEEKS, and MONTHS timespans are updated to retain the latest day, week-to-date, and month-to-date data.

If DAYS-level accounting support is in effect for the DESDKU file, the USRADKU exit is available during this processing phase. This exit is described further in section 9.3 "User Exits."

9.2.4 - File Aging

The files are aged to delete the oldest cycle and rename the work files to be the current cycle.

There are no exits provided in this phase.

9.3 - User Exits

A number of exits are provided to enable the user to add specific record selection and edit checks to the CA MICS VAX/VMS Analyzer without having to modify the Analyzer itself. The CA MICS System Modification Guide (SMG) describes the general principles of exit processing in CA MICS. Refer to section 4.3.2.1 "Definition and Methodology" in the SMG for this description. The VAX/VMS Analyzer employs the Indirect Method described in section 4.3.1.2 "Indirect Exits Points" in this topic area of the SMG.

The remainder of this section describes the exits specific to the VAX/VMS Analyzer.

USRSEL Exit

The USRSEL user exit is invoked immediately after each VMS ACCOUNTING, VMS MONITOR, and CA MICS DEXSUS record has been read. The user may examine the record and indicate whether it is to be kept and processed or discarded. Only a few of the record's fields have been read and decoded at this point. Refer to the source code in:

sharedprefix.MICS.SOURCE(DEXACT, DEXMON, and DEXSUSA),

to determine which data elements are available. For more information on the USRSEL exit, refer to section 4.3.2.1 in the SMG.

USRSfff Exit

The USRSfff user exit is invoked immediately after the data for this record and file has been read and decoded and immediately before the observation is written to the CA MICS work file. All of the record's fields have been read and decoded. Additional edits, computations, and selection checks can be made at this point. Figure 9-2 lists the CA MICS files and their corresponding exit names:

File Name	Exit Name
DEAPRC	USRSPRC
DEAIMG	USRSIMG
DEAINT	USRSINT
DEA_LF	USRS_LF
DEAPRQ	USRSPRQ
DEMPRO	USRSPRO
DEMSTA	USRSSTA
DEMMOD	USRSMOD
DEMPGE	USRSPGE
DEMIOV	USRSIOV
DEMFCP	USRSFCP
DEMPOL	USRSPOL
DEMLOK	USRSLOK
DEMDNT	USRSDNT
DEMFSC	USRSFSC
DEMDSK	USRSDSK
DEMDLK	USRSDLK
DEMSCS	USRSSCS
DEMSYT	USRSSYT
DEMCLS	USRSCLS
DEMRMS	USRSRMS
DEMMSC	USRSMSC
DEMSPR	USRSSPR
DEMTRN	USRSTRN
DEMVPR	USRSVPR
DESDKD	USRSDKD
DESDKQ	USRSDKQ
DESDKU	USRSDKU
DESPRX	USRSPRX
DESSYU	USRSSYU

Figure 9-2. File/User Exit Correspondence

USRAfff Accounting Exit

The USRAfff accounting exit is invoked immediately before the observation is written from the CA MICS work file to the CA MICS DETAIL file. All of the record's fields have been read and decoded. Additional edits, computations, and selection checks can be made at this point. Figure 9-3 lists the CA MICS files and their corresponding accounting exit names:

File Name	Exit Name
DEAPRC	USRAPRC
DEAIMG	USRAIMG
DEA_LF	USRA_LF
DEAPRQ	USRAPRQ
DESDKU *	USRADKU

Figure 9-3. File/Accounting Exit Correspondence

(*) Invocation of the USRADKU Accounting Exit is controlled by the DISKACCT option in DEXOPS. The code to invoke this exit is located in sharedprefix.SOURCE members #DKUDSUM and $DEXMSTR. The DEXPGEN function sets the calling sequence, during either DETAIL or DAYS level processing, based on the corresponding setting in DEXOPS.

9.4 - Using COMMON Input DDnames and DEXINPUT exits

The COMMON statement is an optional DEXOPS parameter entry designed primarily for handling VMS data that has been preprocessed into a special user-defined format for consolidated input of many systems through a single input DDname. There is no standard format for consolidating data from multiple VMS systems, nor is there a standard method. If you wish to consolidate data in this manner, you must define the method and format. If you do so, be sure to preserve original data record order and relative positions of standard data fields when combining files into a consolidated format.

SUGGESTED FORMAT FOR CONSOLIDATED DATA

DEXPGEN supports a suggested format for consolidated data, or you can define your own data format. The suggested format supported by DEXPGEN is as follows:

```
Record Prefix    (14 bytes)   +  Standard Record Contents
-----------------------------    --------------------------
NODENAME 10bytes  TYPE 4bytes    (as provided by VAX/VMS)
```

where:

NODENAME is a 10 byte ASCII field identifying the VAX/VMS system node that is the source of the data.

TYPE is a 4 byte ASCII field identifying the VAX/VMS data source (ACCO for ACCOUNTING, MNTR for MONITOR, and DXSU for DEXSUS).

In this format, every input record is prefixed with 14 bytes of information about system identification and data type. This is followed by the standard record layout for each input record. Support for this format is based on the assumption that the data is consolidated before transmission to MVS and thus it is assumed that the prefix information is recorded in ASCII.

OTHER FORMATS FOR CONSOLIDATED DATA

If you want to use a format that varies from the suggested scheme, it is possible to override these assumptions with your own definition of the DEXINPUT exits (DEXINPTA, DEXINPTM, and DEXINPTS). DEXPGEN supports the suggested format by generating macros to prefix.MICS.USER.SOURCE($DEXMSTR). The code is structured to make it easy for you to override the suggested format if

necessary. Overrides should be defined in the module prefix.MICS.USER.SOURCE($DEXEXIT). For example, to read NODENAME and TYPE as EBCDIC rather than ASCII characters, code the following macro overrides in $DEXEXIT:

```
%MACRO DEXINPTM;
    ROFFSET=14;
    INPUT @1
     VAXID $CHAR10. VAXDS $CHAR4.
     @ ;
    IF VAXDS NE "MNTR" THEN GOTO EOF_CK;
%MEND  DEXINPTM;

%MACRO DEXINPTA;
    ROFFSET=14;
    INPUT @1
     VAXID $CHAR10. VAXDS $CHAR4.
     @ ;
    IF VAXDS NE "ACCO" THEN GOTO EOF_CK;
%MEND  DEXINPTA;

%MACRO DEXINPTS;
    ROFFSET=14;
    INPUT @1
     VAXID $CHAR10. VAXDS $CHAR4.
     @ ;
    IF VAXDS NE "DXSU" THEN GOTO EOF_CK;
%MEND  DEXINPTS;
```

To learn more about how DEXPGEN supports these macros, review the code in sharedprefix.MICS.SOURCE(DEXACT,DEXMON,DEXSUSA) and in prefix.MICS.USER.SOURCE($DEXMSTR).

COMMON STATEMENT

The COMMON statement is used to identify the JCL reference to be used when reading MONITOR, ACCOUNTING, or DEXSUS data for one or more VMS systems from the same input DDname.

```
+-------------------------------------------------------------+
|COMMON datatype ddname EXIT (optional for MONITOR & DEXSUS)|
+-------------------------------------------------------------+
```

where:

'datatype' is MONITOR, ACCOUNT, or DEXSUS.

'ddname' is the DDname that will be used to read this datatype for the systems.

EXIT is a literal keyword that indicates you wish to invoke the special purpose DEXINPUT exit routines for the datatype indicated. EXIT is optional for MONITOR and DEXSUS, and is required for ACCOUNT. This is due to the fact that system identification is NOT possible using ACCOUNTING data without an EXIT, while MONITOR and DEXSUS data normally contains a system identification record.

Sample COMMON Statements:

```
COMMON MONITOR DEXMON  EXIT
COMMON ACCOUNT DEXACC  EXIT
COMMON DEXSUS  DEXSUS  EXIT
```

The definition of the data items described above is made through the DEXOPS member in the prefix.MICS.PARMS library. A sample definition is illustrated below. Note that the 'ddname' field on the MONITOR, ACCOUNT, and DEXSUS statements is specified as '*' to indicate that the COMMON statement will provide the DDname for this data.

SAMPLE DEFINITION USING DEXINPUT EXITS:

```
COMMON MONITOR DEXMON  EXIT
COMMON ACCOUNT DEXACC  EXIT
COMMON DEXSUS  DEXSUS  EXIT
OPTIONS   VMSA  SYSTEMA
  MONITOR VMSA  *
  ACCOUNT VMSA  *
  DEXSUS  VMSA  *
OPTIONS   VMSB  SYSTEMB
  MONITOR VMSB  *
  ACCOUNT VMSB  *
  DEXSUS  VMSB  *
OPTIONS   VMSC  SYSTEMC
  MONITOR VMSC  *
  ACCOUNT VMSC  *
  DEXSUS  VMSC  *
```

Input DD Statements for VAX/VMS (INPUTDEX)

The input DD statements to define the VAX/VMS input files must be supplied in the INPUTDEX member of the prefix.MICS.PARMS library, and are referenced by the MONITOR, ACCOUNT, DEXSUS, and COMMON statements of DEXOPS.

To provide for MONITOR, ACCOUNTING, or DEXSUS data from multiple systems though a single DDname, you must specify a

COMMON statement in DEXOPS for each type of data. You must also provide a DDname in INPUTDEX that corresponds to the DDname on the COMMON statement. This DDname may contain data from one or more VMS systems.

For example, in the JCL below, the unit will process VMS ACCOUNTING data from SYS1 and SYS2, VMS MONITOR data from SYS1, SYS2, and SYS3, and DEXSUS data from SYS2 and SYS3. Each type of data will be brought in using a single DDname, specified in DEXOPS. The DD statements will read as follows:

```
//DEXACC   DD DISP=SHR,DSN=VMS.SYS1.ACCOUNT.DATA
//         DD DISP=SHR,DSN=VMS.SYS2.ACCOUNT.DATA
//DEXMON   DD DISP=SHR,DSN=VMS.SYS1.MONITOR.DATA
//         DD DISP=SHR,DSN=VMS.SYS2.MONITOR.DATA
//         DD DISP=SHR,DSN=VMS.SYS3.MONITOR.DATA
//DEXSUS   DD DISP=SHR,DSN=VMS.SYS2.DEXSUS.DATA
//         DD DISP=SHR,DSN=VMS.SYS3.DEXSUS.DATA
```

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 10

MODIFICATION

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

CHAPTER 10

MODIFICATION

Chapter 10. MODIFICATION

The CA MICS VAX/VMS Analyzer is tailored primarily through the use of options and parameters, file and data element tailoring, and exits. This chapter provides checklists for changing the product's parameters and describes the exits that are available during the DAILY update processing.

Tailoring the VAX/VMS Analyzer is important. If you are using all the defaults shipped with the product, you are probably wasting DASD space. To ensure efficient space utilization, tailoring is required.

The following section is provided:

1 - Parameter Modification

10.1 - Parameter Modification

Numerous parameters control the operation of the VAX/VMS Analyzer. Often, you can customize the product by altering the same parameters that you used to install the product. This section provides information and checklists for modifying the VAX/VMS Analyzer through parameter modification. The following topics are addressed:

1 - Change the Number of Work Files for DAY074
2 - Enable Internal Step Restart
3 - Enable Incremental Update

10.1.1 - Change the Number of Work Files for DAY074

To change the number of work files used in the CA MICS VAX/VMS Analyzer processing in Step DAY074, follow the checklist provided below for each unit.

See Section 7.3.1 of this guide for a detailed description of the statement syntax.

```
*****************************************************
*                                                   *
*           CHANGE NUMBER OF WORK FILES             *
*                                                   *
*****************************************************
```

___ 1. Update the WORK statement in prefix.MICS.PARMS(cccOPS), where (ccc) is the component identifier, to specify the number of work data sets required. Below is an example:

```
WORK n STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss))
```

where:

```
n        - is the number of WORK data sets.
STORCLAS - specifies a storage class for a new data
           set.  The name can have up to eight
           characters.
SPACE    - specifies how much disk space to provide
           for a new data set being allocated.
XXX      - is TRK or CYL.
pppp     - is the primary allocation.
ssss     - is the secondary allocation.
```

You should specify the minimum number of WORK data sets to meet your work space requirements. As a start, try incrementing the number gradually beginning from the default.

___ 2. If this is the first time you are implementing multiple work files for this product, then continue with Step 2. If you are just changing the number currently in use, or simply the space definitions, then proceed to Step 3 of this checklist.

Browse sharedprefix.MICS.PROTOLIB(DYcccnnn) and sharedprefix.MICS.PROTOLIB(cccINCR), where (nnn) is the job step number and (ccc) is the product ID for this product, checking for the presence of the WORK symbolic on the EXEC statement to determine if you have previously modified this product to increase the allocation of SAS WORK space.

___ 2a. If you find a WORK symbolic, simply divide the primary and secondary allocation values from the WORK symbolic by the number of work files specified above (value of n on the WORK statement coded in Step 1).

Coding the resulting values will yield the same aggregate space allocation you have been using with a single WORK file. To double your available WORK space, carry out the division, double the results and use the values in the WORK definition above.

___ 2b. If you did not find a WORK symbolic in PROTOLIB, examine prefix.MICS.PARMS(JCLDEF) for each CA MICS unit containing this product. Find the WORKSPACE keyword. The space allocation specified is used for a single SAS WORK file. Perform the same division as described in the previous paragraph to determine the quantity that will yield equivalent total allocation with multiple WORK files. Then adjust the values upward to meet your needs.

___ 3. Submit the job in prefix.MICS.CNTL(cccPGEN).

___ 4. If you specified RESTART YES in the product's cccOPS, you are done. Otherwise, you must do Steps 5, 6, and 7.

___ 5. Edit prefix.MICS.PARMS(JCLGENU) so that it contains a single line that reads:

DAILY

or, if incremental update is enabled for this product in this unit database, specify:

DAILY INCRccc

where ccc is the product ID.

___ 6. Submit the job in prefix.MICS.CNTL(JCLGENU). Ensure that there are no error messages in MICSLOG or SYSTSPRT, that the MICSLOG contains the normal termination message, BAS10999I, and that the job completes with a condition code of zero.

___ 7. The following operational job(s) have changed:

DAILY

INCRccc (if incremental update is enabled)

If your site has implemented the operational CA MICS processes in a scheduling product, the JCL may have to be refreshed in that product. See the scheduling product's administrator for the exact processes involved in updating that product's representation of the CA MICS jobs.

10.1.2 - Enable Internal Step Restart

To enable the internal step restart in the CA MICS VAX/VMS Analyzer, follow the checklist provided below:

See Section 7.3.1 of this guide for a detailed description of the statement syntax.

```
**************************************************
*                                                *
*          ENABLE INTERNAL STEP RESTART          *
*                                                *
**************************************************
```

___ 1. Edit prefix.MICS.PARMS(cccOPS), where (ccc) is the component identifier, and specify:

RESTART YES

For additional information on related topic, review the documentation for this product on WORK, RESTARTWORK, and RESTARTCKPT parameters to override default data set allocation parameters.

___ 2. Submit the job in prefix.MICS.CNTL(cccPGEN).

___ 3. Edit prefix.MICS.PARMS(JCLGENU) so that it contains a single line that reads:

DAILY

or, if incremental update is enabled for this product in this unit database, specify:

DAILY INCRccc

where ccc is the product ID.

___ 4. Submit the job in prefix.MICS.CNTL(JCLGENU). Ensure that there are no error messages in MICSLOG or SYSTSPRT, that the MICSLOG contains the normal termination message, BAS10999I, and that the job completes with a condition code of zero.

___ 5. The following operational job(s) have changed:

DAILY

INCRccc (if incremental update is enabled)

If your site has implemented the operational CA MICS processes in a scheduling product, the JCL may have

to be refreshed in that product. See the scheduling product's administrator for the exact processes involved in updating that product's representation of the CA MICS jobs.

10.1.3 - Implement Incremental Update

To implement incremental update in the CA MICS VAX/VMS Analyzer, follow the checklist provided below:

See Section 7.3.1 of this guide for a detailed description of the statement syntax.

```
*****************************************************
*                                                   *
*           IMPLEMENT INCREMENTAL UPDATE            *
*                                                   *
*****************************************************
```

___ 1. Edit prefix.MICS.PARMS(cccOPS), where (ccc) is the component identifier:

- o Specify the following:

  INCRUPDATE YES

- o If you want to store the incremental update database files on tape between incremental updates, specify this:

  INCRDB TAPE #gdgs

- o If you want to allocate the incremental update database files during the first incremental update of the day and delete these data sets at the end of the DAILY job step, specify this:

  INCRDB DYNAM

- o If you specified INCRDB TAPE or INCRDB DYNAM, then you must also specify this:

  INCRDETAIL data_set_allocation_parameters
  INCRDAYS data_set_allocation_parameters

- o If you want the incremental update job for this product to get input measurement data from the output of the SPLITSMF job, specify this:

  INCRSPLIT USE data_set_allocation_parameters

- o For additional information on related topic, review the documentation for this product on INCRCKPT, INCRDETAIL, INCRDAYS, or INCRSPLIT parameters to override default data set allocation parameters.

___ 2. Submit the job in prefix.MICS.CNTL(cccPGEN).

___ 3. Edit prefix.MICS.PARMS(JCLGENU) so that it contains two or more lines reading:

```
DAILY
INCRccc cccIUALC cccIUGDG
```

___ 4. Submit the job in prefix.MICS.CNTL(JCLGENU). Ensure that there are no error messages in MICSLOG or SYSTSPRT, that the MICSLOG contains the normal termination message, BAS10999I, and that the job completes with a condition code of zero.

___ 5. Edit the job in prefix.MICS.CNTL(cccIUALC).

- o Inspect and/or specify data set allocation parameters for the incremental update database and checkpoint files. If you specified INCRDB TAPE or INCRDB DYNAM, the cccIUALC job will only allocate the incremental update checkpoint data set.

- o Submit the job. Ensure that there are no error messages in MICSLOG or SASLOG, and that the job completes with a condition code of zero.

___ 6. If you specified INCRDB TAPE, submit the job in prefix.MICS.CNTL(cccIUGDG) to define generation group indexes for the incremental update DETAIL and DAYS tape data sets. Examine SASLOG, MICSLOG, and SYSPRINT to verify that the generation group indexes were correctly defined.

Note: You may see error messages for the DLTX (or DELETE) statements. This is not a problem. cccIUGDG deletes each index prior to defining it, and an error message is issued if the index does not yet exist (e.g., if this is the first time you ran the cccIUGDG job).

___ 7. The following operational job(s) have changed:

DAILY INCRccc

If your site has implemented the operational CA MICS processes in a scheduling product, the JCL may have to be refreshed in that product. See the scheduling product's administrator for the exact processes involved in updating that product's representation of the CA MICS jobs.

___ 8. Implement operational procedures for gathering input measurement data and executing incremental updates (INCRccc) during the day.

You may also need to modify operational procedures for the DAILY job to ensure that processing is limited to input measurement data that has not been input to one of the day's incremental update executions.

SPECIAL CONSIDERATIONS:

There are a number of files in the VAX/VMS Analyzer database that contain metrics which are the result of "delta" calculations done from one interval to the next. In these files, the first interval encountered on any given processing run is used to initialize the values for the subsequent "delta" calculations. Because of this, that first time interval in most cases will NOT be represented in the output file.

This architecture is acceptable as it only comes into play once for each DAILY processing cycle. When the Incremental Update feature is activated, this condition may become more noticeable, depending on the number of increments in any given DAILY processing cycle.

It is important to schedule your incremental cutoffs at off peak times to avoid the reporting impacts of a missing interval.

The following VAX/VMS Analyzer files have been identified as being susceptible to the condition described above:

```
DEMDSK - VMS MONITOR Disk File
DEMMOD - VMS MONITOR Modes File
DEMPRO - VMS MONITOR Process File
DEMSCS - VMS MONITOR SCS File
DEMSPR - VMS System Profile File
DESPRX - VMS Process Activity File
```

Another important area of consideration is the processing of DEXSUS data. DEXSUS data is created by running, once a day, a script of commands on a VAX machine, which takes a snapshot of the system environment. Since this is a once a day activity, the DEXSUS data will only be processed in the DAILY processing step DAY074. The DEXSUS data will not be processed in the INCRDEX job even though the DD for the data is present.

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

APPENDIX A

MESSAGES

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

APPENDIX A

MESSAGES

APPENDIX A. MESSAGES

This appendix lists all messages generated by the CA MICS VAX/VMS Analyzer. Some messages are generated during the processing of the control statements, while others are caused by various conditions in the data found during the processing. The messages are listed in ascending numerical sequence and include the full text of the message, the type, the reason for the messages, appropriate user action, and applicable references to documentation.

The following type codes are used to categorize the messages:

Information	designates a note that documents a CA MICS VAX/VMS Analyzer option or potentially important feature in the data.
Warning	designates a condition in either the data or control statements that does not affect the CA MICS VAX/VMS Analyzer's operation but that may lead to unexpected results.
Error	designates that a problem has been encountered with a control statement that will prevent a successful run of the CA MICS VAX/VMS Analyzer. Execution is stopped after all control statements are processed.

The message text often includes references to information that is contained in a control statement or values of permanent or temporary variables created during CA MICS VAX/VMS Analyzer processing. In the description below, the message text contains a word beginning with a percent sign (%), which indicates that a value will be substituted into the text at execution time.

DEX0001

TEXT:	INPUT: DSN=%X VOL=SER=%PVOL
TYPE:	Information
REASON:	Logs the name of each input parameter read by the DEXPGEN process.
ACTION:	None
REFERENCES:	Section 7.3.3

DEX0002

TEXT:	DEXOPS OPTIONS SYSTEM ID: %KEYWORD2 IS MORE THAN 4 CHARACTERS IN LENGTH
TYPE:	Error
REASON:	The indicated SYSID on the DEXOPS OPTIONS statement was longer than the 4-character maximum allowed.
ACTION:	Shorten the SYSID name on the OPTIONS statement in prefix.MICS.PARMS(DEXOPS).
REFERENCES:	Section 7.3.1

DEX0003

TEXT:	SYSID PARM CPU MODEL NUMBER: %KEYWORD3 IS MORE THAN 4 CHARACTERS IN LENGTH
TYPE:	Warning
REASON:	The indicated CPU model number in SYSID was longer than the 4-character maximum expected.
ACTION:	Check the CPU model number on the statement in prefix.MICS.PARMS(SYSID).
REFERENCES:	- Section 7.4 - PIOM Guide 2.3.2.2 - see also Administrative Newsletter, "Guidelines for Coding SYSID Parms"

```
+---------+
| DEX0004 |
+---------+

   TEXT:         DEXOPS DEFAULT DURATION VALUE:  %KEYWORD4 IS
                 NOT A POSITIVE NUMERIC VALUE, AS REQUIRED.

   TYPE:         Error

   REASON:       The default duration value for a DEXSUS data
                 statement is not a valid numeric value or is
                 missing entirely.

   ACTION:       Correct the default duration value on the
                 DEXSUS statement in
                 prefix.MICS.PARMS(DEXOPS).

   REFERENCES:   Section 7.3.1

+---------+
| DEX0005 |
+---------+

   TEXT:         SCP SPECIFIED IN SYSID FOR %KEYWORD2 IS NOT
                 A VAX/VMS SYSTEM.
                 VALID VALUES ARE *, VMS, OR VAXVMS.

   TYPE:         Warning

   REASON:       The SCP name in SYSID is not specified
                 correctly or the field is missing entirely.

   ACTION:       Provide a recognized VAX/VMS SCP name on
                 the statement in prefix.MICS.PARMS(SYSID).

   REFERENCES:   - Section 7.4
                 - PIOM Guide 2.3.2.2
                 - see also Administrative Newsletter,
                  "Guidelines for Coding SYSID Parms"

+---------+
| DEX0006 |
+---------+

   TEXT:         DEXOPS NODE NAME MISSING FOR %KEYWORD2 ENTRY

   TYPE:         Error

   REASON:       The indicated entry on the DEXOPS OPTIONS
                 statement was missing a node name entry to
                 map to the ORGSYSID listed as %KEYWORD2.

   ACTION:       Specify the VAX node name that is to be
                 mapped to the ORGSYSID, on the OPTIONS
                 statement in prefix.MICS.PARMS(DEXOPS).

   REFERENCES:   Section 7.3.1

+---------+
| DEX0007 |
+---------+

   TEXT:         MORE THAN ONE OPTIONS STATEMENT FOR ORGSYSID
                 %KEYWORD2 FOUND IN DEXOPS

   TYPE:         Error

   REASON:       More than one OPTIONS statement was included
                 in the DEXOPS member for the same ORGSYSID.

   ACTION:       Review prefix.MICS.PARMS(DEXOPS) and correct
                 or remove the inappropriate OPTIONS
                 statement.

   REFERENCES:   Section 7.3.1
```

```
+---------+
| DEX0008 |
+---------+

   TEXT:         INVALID NUMERIC DATA:  %CVAL

   TYPE:         Error

   REASON:       Value in parameter line is not a valid
                 numeric value.

   ACTION:       Review prefix.MICS.PARMS(DEXOPS) and correct
                 the indicated statement.

   REFERENCES:   Section 7.3.1


+---------+
| DEX0010 |
+---------+

   TEXT:         ORIGINAL SYSID:   %KEYWORD1 HAS BEEN
                 PREVIOUSLY DEFINED.

   TYPE:         Error

   REASON:       The ORGSYSID has been defined more than once
                 in the SYSID parameter.

   ACTION:       Review prefix.MICS.PARMS(SYSID) and correct
                 the inappropriate ORGSYSID definition.

   REFERENCES:   - Section 7.4
                 - PIOM Guide 2.3.2.2
                 - see also Administrative Newsletter,
                   "Guidelines for Coding SYSID Parms"


+---------+
| DEX0011 |
+---------+

   TEXT:         THE MAXIMUM SYSID LIMIT HAS BEEN REACHED.
                 PLEASE CONTACT CA MICS PRODUCT SUPPORT.

   TYPE:         Error

   REASON:       More than 100 entries were specified in
                 prefix.MICS.PARMS(SYSID).

   ACTION:       Contact CA MICS Product Support.

   REFERENCES:   - Section 7.4
                 - PIOM Guide 2.3.2.2
                 - see also Administrative Newsletter,
                   "Guidelines for Coding SYSID Parms"


+---------+
| DEX0012 |
+---------+

   TEXT:         SPECIFIED DDNAME:  %KEYWORD3 IS MORE THAN 8
                 CHARACTERS IN LENGTH.

   TYPE:         Error

   REASON:       DDnames cannot be more than 8 characters in
                 length.

   ACTION:       Reduce the length of the DDname on the
                 statement indicated (MONITOR, ACCOUNT, or
                 COMMON) in prefix.MICS.PARMS(DEXOPS).  Also
                 ensure that the INPUTDEX PARMS member
                 specifies a corresponding DD statement.

   REFERENCES:   Sections 7.3.1 and 7.3.2
```

```
+---------+
| DEX0013 |
+---------+

   TEXT:         THIS ORGSYSID:  %KEYWORD2 AND DATA SOURCE:
                 %KEYWORD1 HAVE ALREADY BEEN DEFINED.

   TYPE:         Error

   REASON:       For a given ORGSYSID, only one MONITOR and
                 one ACCOUNT statement may be defined.

   ACTION:       Review prefix.MICS.PARMS(DEXOPS) and correct
                 the inappropriate MONITOR or ACCOUNT
                 statement so that you are not attempting to
                 assign two different input DDnames to the
                 same ORGSYSID for the same data type.

   REFERENCES:   Section 7.3.1

+---------+
| DEX0014 |
+---------+

   TEXT:         SPECIFIED ORGSYSID:  %KEYWORD2 HAS NOT BEEN
                 DEFINED ON A PREVIOUS OPTION STATEMENT

   TYPE:         Error

   REASON:       The ORGSYSID named on the MONITOR or ACCOUNT
                 statement must first be defined on an
                 OPTIONS statement.

   ACTION:       Review prefix.MICS.PARMS(DEXOPS) and add
                 correct the appropriate OPTIONS statement.

   REFERENCES:   Section 7.3.1

+---------+
| DEX0015 |
+---------+

   TEXT:         SUBPARAMETER:  %KEYWORD2 IS NOT VALID ON A
                 %KEYWORD1 STATEMENT

   TYPE:         Error

   REASON:       The subparameter is not a recognized entry
                 for the indicated statement.

   ACTION:       Review prefix.MICS.PARMS(DEXOPS) and correct
                 the indicated statement.

   REFERENCES:   Section 7.3.1

+---------+
| DEX0016 |
+---------+

   TEXT:         KEYWORD:  %KEYWORD1 IS NOT RECOGNIZED

   TYPE:         Error

   REASON:       The keyword statement is not a recognized
                 entry for DEXOPS.

   ACTION:       Review prefix.MICS.PARMS(DEXOPS) and correct
                 the indicated statement.

   REFERENCES:   Section 7.3.1

+---------+
| DEX0017 |
+---------+

   TEXT:         END OF MEMBER %PMEM INPUT

   TYPE:         Information

   REASON:       Indicates end of each input parameter read
                 by the DEXPGEN process.

   ACTION:       None

   REFERENCES:   Section 7.3.3
```

```
+---------+
| DEX0018 |
+---------+

   TEXT:         REQUIRED STATEMENT MISSING:  %STYP

   TYPE:         Error

   REASON:       The keyword statement is a required entry
                 for DEXOPS.

   ACTION:       Review prefix.MICS.PARMS(DEXOPS) and supply
                 the required statement.

   REFERENCES:   Section 7.3.1

+---------+
| DEX0019 |
+---------+

   TEXT:         ERROR(S) ENCOUNTERED IN DEXOPS UNIT
                 PARAMETER DEFINITIONS, MACRO GENERATION
                 ABORTED.  CORRECT THE ERROR(S), AND
                 RE-EXECUTE DEXPGEN

   TYPE:         Error

   REASON:       At least one fatal error was encountered
                 while processing DEXOPS.

   ACTION:       Correct the error(s) and resubmit the
                 DEXPGEN job.

   REFERENCES:   Section 7.3.1, 7.3.2, and 7.3.3

+---------+
| DEX0020 |
+---------+

   TEXT:         DUPLICATE DDNAMES NOT ALLOWED:  %KEYWORD3

   TYPE:         Error

   REASON:       The DDname has already been specified for
                 use with another MONITOR or ACCOUNT input
                 source.

   ACTION:       Review prefix.MICS.PARMS(DEXOPS) and supply
                 a unique DDname for use with this input data
                 source.

   REFERENCES:   Sections 7.3.1 and 7.3.2

+---------+
| DEX0021 |
+---------+

   TEXT:         COMMON INPUT DD NOT ALLOWED FOR:  %KEYWORD1

   TYPE:         Error

   REASON:       COMMON input requested a data source (e.g.,
                 ACCOUNT) that does not contain system
                 identification information.  COMMON cannot
                 support this data source because SYSID can
                 not be determined.

   ACTION:       Supply a unique DDname in the indicated
                 statement in prefix.MICS.PARMS(DEXOPS), and
                 provide a corresponding DD statement in
                 prefix.MICS.PARMS(INPUTDEX).

   REFERENCES:   Sections 7.3.1, 7.3.2, and 9.4
```

```
+---------+
| DEX0022 |
+---------+

   TEXT:         COMMON INPUT DD FOR %KEYWORD1 HAS ALREADY
                 BEEN DEFINED.

   TYPE:         Error

   REASON:       Only one common DD is supported for a given
                 data type (i.e., MONITOR).

   ACTION:       Use the common DD already defined or supply
                 a unique DDname in the indicated statement
                 in prefix.MICS.PARMS(DEXOPS).  Check the
                 corresponding DD statements in
                 prefix.MICS.PARMS(INPUTDEX).

   REFERENCES:   Sections 7.3.1 and 7.3.2

+---------+
| DEX0023 |
+---------+

   TEXT:         MEMBER %PMEM IS EMPTY OR CONTAINS NO DD
                 STATEMENTS

   TYPE:         Error

   REASON:       No active DD statements were found in the
                 indicated member in prefix.MICS.PARMS.  The
                 DDnames used in DEXOPS require matching DD
                 statements in the input JCL.

   ACTION:       Review prefix.MICS.PARMS member and supply
                 the appropriate DD statements.

   REFERENCES:   Section 7.3.2

+---------+
| DEX0024 |
+---------+

   TEXT:         DDNAME %KEYWORD3 NOT FOUND IN INPUTDEX

   TYPE:         Error

   REASON:       No matching DD statement was found in
                 prefix.MICS.PARMS(INPUTDEX).  The DDnames
                 used in DEXOPS require matching DD
                 statements in the input JCL.

   ACTION:       Review prefix.MICS.PARMS(INPUTDEX) and
                 prefix.MICS.PARMS(DEXOPS) and correct the
                 discrepancy in the use of DDnames.

   REFERENCES:   Sections 7.3.1 and 7.3.2

+---------+
| DEX0025 |
+---------+

   TEXT:         INPUTDEX DDNAME:  %Y NOT REFERENCED IN
                 DEXOPS

   TYPE:         Warning

   REASON:       DDnames are not processed by the DAILY job
                 unless they are specified for use in
                 prefix.MICS.PARMS(DEXOPS).

   ACTION:       Check prefix.MICS.PARMS(INPUTDEX) to ensure
                 that the presence of the unreferenced DDname
                 is intentional.

   REFERENCES:   Section 7.3.2
```

```
+---------+
| DEX0029 |
+---------+

   TEXT:         INPUTDEX DDNAME:  %X IS MORE THAN 8
                 CHARACTERS IN LENGTH.

   TYPE:         Error

   REASON:       DDnames longer than 8 characters will cause
                 the DAILY job to end with a JCL error.

   ACTION:       Correct the invalid DDname in
                 prefix.MICS.PARMS(INPUTDEX).

   REFERENCES:   Section 7.3.2

+---------+
| DEX0031 |
+---------+

   TEXT:         ORGSYSID %KEYWORD2 IS NOT DEFINED IN
                 PARMS(SYSID).

   TYPE:         Error

   REASON:       The ORGSYSID has not been defined in the
                 SYSID parameter, or BASPGEN may need to be
                 run.

   ACTION:       Review prefix.MICS.PARMS(SYSID) and add the
                 appropriate ORGSYSID definition if needed.
                 Run BASPGEN and then re-run DEXPGEN (or
                 simply run ALLPGEN).

   REFERENCES:   - Section 7.4
                 - PIOM Guide 2.3.2.2
                 - see also Administrative Newsletter,
                   "Guidelines for Coding SYSID Parms"
```

```
+---------+
| DEX0032 |
+---------+

   TEXT:         MORE THAN ONE DISKACCT STATEMENT FOUND IN
                 DEXOPS.

   TYPE:         Error

   REASON:       No more than one DISKACCT statement should
                 be specified in the DEXOPS member.  This
                 option applies to the entire unit and cannot
                 be specified separately for different input
                 systems.

   ACTION:       Review prefix.MICS.PARMS(DEXOPS) and remove
                 the inappropriate DISKACCT statement.

   REFERENCES:   Section 7.3.1
```

```
+-----------------------------------------------------------------------------------------------------------------------------------+
|                                                                                                                                   |
| +---------+                                                                                                                       |
| | DEX0041 |                                                                                                                       |
| +---------+                                                                                                                       |
|                                                                                                                                   |
|    TEXT:         >>> START OF DEXACT MODULE PROCESSING <<<                                                                        |
|                                                                                                                                   |
|    TYPE:         Information                                                                                                      |
|                                                                                                                                   |
|    REASON:       Indicates start of input format processing                                                                       |
|                  for VMS ACCOUNTING data.                                                                                         |
|                                                                                                                                   |
|    ACTION:       None                                                                                                             |
|                                                                                                                                   |
|    REFERENCES:   Section 9.2.2                                                                                                    |
|                                                                                                                                   |
| +---------+                                                                                                                       |
| | DEX0042 |                                                                                                                       |
| +---------+                                                                                                                       |
|                                                                                                                                   |
|    TEXT:         >>> END OF DEXACT MODULE PROCESSING <<<                                                                          |
|                                                                                                                                   |
|    TYPE:         Information                                                                                                      |
|                                                                                                                                   |
|    REASON:       Indicates end of input format processing for                                                                     |
|                  VMS ACCOUNTING data.                                                                                             |
|                                                                                                                                   |
|    ACTION:       None                                                                                                             |
|                                                                                                                                   |
|    REFERENCES:   Section 9.2.2                                                                                                    |
|                                                                                                                                   |
| +---------+                                                                                                                       |
| | DEX0043 |                                                                                                                       |
| +---------+                                                                                                                       |
|                                                                                                                                   |
|    TEXT:         >>> START OF DEXSUSA MODULE PROCESSING <<<                                                                       |
|                                                                                                                                   |
|    TYPE:         Information                                                                                                      |
|                                                                                                                                   |
|    REASON:       Indicates start of input format processing                                                                       |
|                  for CA MICS DEXSUS data.                                                                                         |
|                                                                                                                                   |
|    ACTION:       None                                                                                                             |
|                                                                                                                                   |
|    REFERENCES:   Section 9.2.2                                                                                                    |
| +---------+                                                                                                                       |
| | DEX0044 |                                                                                                                       |
| +---------+                                                                                                                       |
|                                                                                                                                   |
|    TEXT:         >>> END OF DEXSUSA MODULE PROCESSING <<<                                                                         |
|                                                                                                                                   |
|    TYPE:         Information                                                                                                      |
|                                                                                                                                   |
|    REASON:       Indicates end of input format processing for                                                                     |
|                  CA MICS DEXSUS data.                                                                                             |
|                                                                                                                                   |
|    ACTION:       None                                                                                                             |
|                                                                                                                                   |
|    REFERENCES:   Section 9.2.2                                                                                                    |
| +---------+                                                                                                                       |
| | DEX0046 |                                                                                                                       |
| +---------+                                                                                                                       |
|                                                                                                                                   |
|    TEXT:         >>> START OF DEXMON MODULE PROCESSING <<<                                                                        |
|                                                                                                                                   |
|    TYPE:         Information                                                                                                      |
|                                                                                                                                   |
|    REASON:       Indicates start of input format processing                                                                       |
|                  for VMS MONITOR data.                                                                                            |
|                                                                                                                                   |
|    ACTION:       None                                                                                                             |
|                                                                                                                                   |
|    REFERENCES:   Section 9.2.2                                                                                                    |
| +---------+                                                                                                                       |
| | DEX0047 |                                                                                                                       |
| +---------+                                                                                                                       |
|                                                                                                                                   |
|    TEXT:         >>> END OF DEXMON MODULE PROCESSING <<<                                                                          |
|                                                                                                                                   |
|    TYPE:         Information                                                                                                      |
|                                                                                                                                   |
|    REASON:       Indicates end of input format processing for                                                                     |
|                  VMS MONITOR data.                                                                                                |
|                                                                                                                                   |
|    ACTION:       None                                                                                                             |
|                                                                                                                                   |
|    REFERENCES:   Section 9.2.2                                                                                                    |
|                                                                                                                                   |
|                                                                                                                                   |
|                                                                                                                                   
```

```
+---------+
| DEX0054 |
+---------+

   TEXT:         ORGSYSID= %ORGSYSID DDNAME= %DDNAME DATASET
                 NAME= %DSNAME

   TYPE:         Information

   REASON:       Describe each input file processed from the
                 VMS ACCOUNTING data source.  Message is
                 issued before first input from each data
                 set.

   ACTION:       None

   REFERENCES:   None

+---------+
| DEX0055 |
+---------+

   TEXT:         START OF INPUT FROM DDNAME= %DDNAME DATASET
                 NAME= %DSNAME

   TYPE:         Information

   REASON:       Describe each input file processed from the
                 VMS MONITOR and CA MICS DEXSUS data sources.
                 Message is issued before first input from
                 each data set.

   ACTION:       None

   REFERENCES:   None

+---------+
| DEX0056 |
+---------+

   TEXT:         ORGSYSID CANNOT BE DETERMINED.  RUN ABORTED

   TYPE:         Error

   REASON:       Issued during VMS MONITOR data source input
                 processing when SYSTEM IDENTIFICATION record
                 (type 129) is not present.  Also issued
                 during CA MICS DEXSUS data source input
                 processing when SYSTEM INFORMATION records
                 (type 01) are not present.  Indicates an
                 incomplete and possibly erroneous input data
                 file.  May also be issued from VMS
                 ACCOUNTING input if DEXOPS COMMON EXIT code
                 is in effect.

   ACTION:       Locate missing records, or reconstruct the
                 input file.  If you are using COMMON EXITS
                 in DEXOPS, check the logic that assigns
                 VAXID and ORGSYSID in your exit code.

   REFERENCES:   Sections 6.3.2, 6.4.2, 9.2.1, and 9.4

+---------+
| DEX0057 |
+---------+

   TEXT:         TYPE 128 RECORD READ.  RECORD WILL NOT BE
                 KEPT.  MONITOR START= %WRK_DTM1 DURATION=
                 %DEXDUR

   TYPE:         Information

   REASON:       Provide the file header information for each
                 input file processed from the VMS MONITOR
                 data source.  Record is discarded and
                 therefore not included in further record
                 counts.

   ACTION:       None

   REFERENCES:   Sections 6.3.1 and 9.2.1
```

DEX0058

TEXT: NODE NAME: %NODID FROM PARMS(DEXOPS) DOES NOT MATCH NODE NAME: %ZORGID IN MONITOR 129 REC. DEXPGEN MAY BE REQUIRED. OPTION VERIFY=ABORT SPECIFIED IN DEXOPS.

TYPE: Error

REASON: Cannot determine correct ORGSYSID to assign to the data because the VAX node name found in the input data does not agree with node name information generated for the input DDname from DEXOPS.

ACTION: Check the VAX node name that is to be mapped to the ORGSYSID on the OPTIONS statement in prefix.MICS.PARMS(DEXOPS). Compare to the node information given in this message. Determine if discrepancy is due to wrong input file, incorrect DEXOPS specification, or requirement for DEXPGEN.

If you are using COMMON EXITS in the DEXOPS parameter, check the logic that assigns (reads) VAXID in your exit code. You can recognize this case if the message indicates a node name of *DEXINPM* from PARMS(DEXOPS).

REFERENCES: Sections 6.3.2, 7.3.1, 7.3.3, and 9.4

DEX0059

TEXT: NODE NAME: %NODID FROM PARMS(DEXOPS) DOES NOT MATCH NODE NAME: %ZORGID IN MONITOR 129 REC. DEXPGEN MAY BE REQUIRED. OPTION VERIFY=NOABORT SPECIFIED IN DEXOPS.

TYPE: Warning

REASON: Cannot determine correct ORGSYSID to assign to the data because the VAX node name found in the input data does not agree with node name information generated for the input DDname from DEXOPS.

ACTION: Check the VAX node name that is to be mapped to the ORGSYSID on the OPTIONS statement in prefix.MICS.PARMS(DEXOPS). Compare to the node information given in this message. Determine if discrepancy due to wrong input file, incorrect DEXOPS specification, or requirement for DEXPGEN.

If you are using COMMON EXITS in the DEXOPS parameter, check the logic that assigns (reads) VAXID in your exit code. You can recognize this case if the message indicates a node name of *DEXINPM* from PARMS(DEXOPS).

REFERENCES: Sections 6.3.2, 7.3.1, 7.3.3, and 9.4

```
+---------+
| DEX0060 |
+---------+

   TEXT:         TYPE 129 RECORD READ.  RECORD WILL NOT BE
                 KEPT.  ORGSYSID= %ORGSYSID IDENTIFIES NODE=
                 %ZORGID SYSTEM BOOT TIME= %WRK_DTM1

   TYPE:         Information

   REASON:       Provide the SYSID information for each input
                 file processed from the VMS MONITOR data
                 source.  Record is discarded and therefore
                 not included in further record counts.

   ACTION:       None

   REFERENCES:   Sections 6.3.2 and 9.2.1

+---------+
| DEX0061 |
+---------+

   TEXT:         ORGSYSID= %ORGSYSID
                 FIRST INPUT TIME= %WRK_DTM1'

   TYPE:         Information

   REASON:       Provide the SYSID and TIMESTAMP information
                 for the first measurement data record of
                 each input file processed from the VMS
                 MONITOR data source.

   ACTION:       None

   REFERENCES:   Section 9.2.1

+---------+
| DEX0062 |
+---------+

   TEXT:         TYPE 129 RECORD FOUND BEFORE TYPE 128
                 RECORD.  UNEXPECTED RECORD SEQUENCE.  CHECK
                 INPUT DATA.

   TYPE:         Error

   REASON:       Implies that VMS MONITOR version and other
                 header record (type 128) data will not be
                 available to the CA MICS VAX/VMS Analyzer.
                 Because record layouts are unknown for an
                 unrecognized version, processing is aborted.

   ACTION:       Ensure that the correct input file has been
                 provided, and if so, contact CA MICS Product
                 Support.

   REFERENCES:   Sections 6.3.1 and 9.2.1

+---------+
| DEX0063 |
+---------+

   TEXT:         NO TYPE 129 RECORD FOUND FOLLOWING TYPE 128
                 UNEXPECTED RECORD SEQUENCE. CHECK INPUT
                 DATA.

   TYPE:         Error

   REASON:       Implies that VMS MONITOR System Information
                 header record (Type 129) data will not be
                 available to the CA MICS VAX/VMS Analyzer.
                 Because this record is needed to tie the
                 nodeid to the sysid, processing is aborted.

   ACTION:       Ensure that the correct input file has been
                 provided, and if so, contact CA MICS Product
                 Support.

   REFERENCES:   Sections 6.3.2 and 9.2.1
```

```
+---------+
| DEX0066 |
+---------+

   TEXT:         UNKNOWN MONITOR VERSION:  %ZZMONVR , RUN
                 ABORTED.

   TYPE:         Error

   REASON:       Indicates a VMS MONITOR version found in a
                 header record (type 128) that is not
                 recognized by the CA MICS VAX/VMS Analyzer.
                 Because record layouts are unknown for an
                 unrecognized version, processing is aborted.

   ACTION:       Ensure that the correct input file has been
                 provided, and if so, contact CA MICS Product
                 Support.

   REFERENCES:   Sections 6.3.1 and 9.2.1


+---------+
| DEX0071 |
+---------+

   TEXT:         TYPE 00 RECORD READ. RECORD WILL NOT BE
                 KEPT.  DEXSUS START= %WRK_DTM1 MAINTLVL=
                 %DEXTMPML

   TYPE:         Information

   REASON:       Provide the file header information for each
                 input file processed from the CA MICS DEXSUS
                 data source.  The record is discarded and
                 therefore not included in further record
                 counts.

   ACTION:       None

   REFERENCES:   Section 6.4.1
```

```
+---------+
| DEX0072 |
+---------+

   TEXT:         UNKNOWN MAINTLVL:  %DEXTMPML , RUN ABORTED.

   TYPE:         Error

   REASON:       Indicates a CA MICS DEXSUS version and
                 maintenance level found in a header record
                 (type 00) that is not recognized by the
                 CA MICS VAX/VMS Analyzer.  Because record
                 layouts are unknown for an unrecognized
                 version, processing is aborted.

   ACTION:       Ensure that the correct input file has been
                 provided, and if so, contact CA MICS Product
                 Support.

   REFERENCES:   Sections 6.4.1 and 9.2.1


+---------+
| DEX0073 |
+---------+

   TEXT:         UNKNOWN VMS VERSION: %SYUVMSVR, RUN ABORTED.

   TYPE:         Error

   REASON:       Indicates the VMS version found in the first
                 data record following a system information
                 separator record (type 01) does not match
                 any VMS version recognized by the CA MICS
                 VAX/VMS Analyzer.  Because record layouts
                 are unknown for an unrecognized version,
                 processing is aborted.

   ACTION:       Ensure that the correct input file has been
                 provided, and if so, contact CA MICS Product
                 Support.

   REFERENCES:   Sections 6.4.2
```

DEX0074

TEXT: NODE NAME: %NODID FROM PARMS(DEXOPS) DOES NOT MATCH NODE NAME: %DEXNODE IN DEXSUS 01-VAX/VMS REC. DEXPGEN MAY BE REQUIRED. OPTION VERIFY=ABORT SPECIFIED IN DEXOPS.

TYPE: Error

REASON: Cannot determine correct ORGSYSID to assign to the data because the VAX node name found in the input data does not agree with node name information generated for the input DDname from DEXOPS.

ACTION: Check the VAX node name that is to be mapped to the ORGSYSID on the OPTIONS statement in prefix.MICS.PARMS(DEXOPS). Compare to the node information given in this message. Determine if the discrepancy is due to the wrong input file, incorrect DEXOPS specification, or requirement for DEXPGEN.

If you are using COMMON EXITS in the DEXOPS parameter, check the logic that assigns (reads) VAXID in your exit code. You can recognize this case if the message indicates a node name of *DEXINPS* from PARMS(DEXOPS).

REFERENCES: Sections 6.4.2, 7.3.1, 7.3.3, and 9.4

DEX0075

TEXT: NODE NAME: %NODID FROM PARMS(DEXOPS) DOES NOT MATCH NODE NAME: %DEXNODE IN DEXSUS 01-VAX/VMS REC. DEXPGEN MAY BE REQUIRED. OPTION VERIFY=NOABORT SPECIFIED IN DEXOPS.

TYPE: Warning

REASON: Cannot determine correct ORGSYSID to assign to the data because the VAX node name found in the input data does not agree with node name information generated for the input DDname from DEXOPS.

ACTION: Check the VAX node name that is to be mapped to the ORGSYSID on the OPTIONS statement in prefix.MICS.PARMS(DEXOPS). Compare to the node information given in this message. Determine if discrepancy due to wrong input file, incorrect DEXOPS specification, or requirement for DEXPGEN.

If you are using COMMON EXITS in the DEXOPS parameter, check the logic that assigns (reads) VAXID in your exit code. You can recognize this case if the message indicates a node name of *DEXINPS* from PARMS(DEXOPS).

REFERENCES: Sections 6.4.2, 7.3.1, 7.3.3, and 9.4

```
+---------+
| DEX0076 |
+---------+

   TEXT:         TYPE 01 RECORDS READ.  RECORDS WILL NOT BE
                 KEPT.
                 ORGSYSID= %ORGSYSID IDENTIFIES NODE=
                 %DEXNODE
                 SHOW SYSTEM RECORDS TIME= %WRK_DTM1

   TYPE:         Information

   REASON:       Provide the SYSID information for each input
                 file processed from the VMS MONITOR data
                 source.  The record is discarded and
                 therefore not included in further record
                 counts.

   ACTION:       None

   REFERENCES:   Sections 6.4.2

+---------+
| DEX0077 |
+---------+

   TEXT:         TYPE 01 RECORD FOUND BEFORE TYPE 00 RECORD.
                 UNEXPECTED RECORD SEQUENCE. CHECK INPUT
                 DATA.

   TYPE:         Error

   REASON:       Implies that DEXSUS version and maintenance
                 level and other data found in a header
                 record header record (type 00) data will not
                 be available to the CA MICS VAX/VMS
                 Analyzer.  Because record layouts are
                 unknown for an unrecognized version,
                 processing is aborted.

   ACTION:       Ensure that the correct input file has been
                 provided, and if so, contact CA MICS Product
                 Support.

   REFERENCES:   Section 6.4.1
```

```
+---------+
| DEX0078 |
+---------+

   TEXT:         SYSTEM BOOT TIME= %WRK_DTM1

   TYPE:         Information

   REASON:       Provide the BOOT TIME information from the
                 type 13 separator record of each input file
                 processed from the CA MICS DEXSUS data
                 source.

   ACTION:       None

   REFERENCES:   Section 6.4.5

+---------+
| DEX0079 |
+---------+

   TEXT:         TYPE 99 RECORD READ. DEXSUS FILE EOF RECORD.
                 DEXSUS END= %WRK_DTM1

   TYPE:         Information

   REASON:       Provide the file trailer information for
                 each input file processed from the CA MICS
                 DEXSUS data source.  When this record is
                 encountered, the input format routine writes
                 a record to the System Status File (DESSYU),
                 and therefore the record is included in step
                 record counts.

   ACTION:       None

   REFERENCES:   Section 6.4.12
```

```
+---------+
| DEX0080 |
+---------+

   TEXT:         NODE NAME:  %DEXTMPNN FROM SYS$NODE DOES NOT
                 MATCH NODE NAME:  %DEXNODE FROM SHOW SYSTEM.

   TYPE:         Warning

   REASON:       Conflicting information has been identified
                 in the DEXSUS type 01 records identifying
                 the VAX node name.  This is the result of a
                 comparison of the SHOW SYSTEM node name
                 output field to the F$TRNLNM("NODENAME")
                 lexical function.  The SHOW SYSTEM value
                 will be assumed unless it is blank.

   ACTION:       Check the VAX node name that is to be mapped
                 to the ORGSYSID on the OPTIONS statement in
                 prefix.MICS.PARMS(DEXOPS).  Compare to the
                 node information given in this message.  If
                 an error situation arises, this message will
                 be followed by other messages with specific
                 corrective actions.

   REFERENCES:   Section 6.4.2
```

```
+---------+
| DEX0081 |
+---------+

   TEXT:         >>>>> DISKACCT OPTION IN EFFECT:  %DEXXACT

   TYPE:         Information

   REASON:       Provide the DISKACCT setting information in
                 effect for processing the DESDKU file data
                 into accounting journal file records.  The
                 value for %DEXXACT should be either DETAIL
                 or DAYS.  This setting originates from the
                 DEXOPS member or is set by default to DETAIL
                 if not specified in DEXOPS.

   ACTION:       None

   REFERENCES:   Sections 2.2.1 and 7.3.1
```

```
+---------+
| DEX0082 |
+---------+

   TEXT:         >>>>> ERROR:  DISKACCT SETTING IS INVALID

   TYPE:         Error

   REASON:       Indicates that the value for the DISKACCT
                 setting was neither DETAIL or DAYS.  This
                 setting originates from the DEXOPS member or
                 is set by default to DETAIL if not specified
                 in DEXOPS.  Because the intended processing
                 path is unknown for an unrecognized setting,
                 and because this will impact the journal
                 entry recording for DESDKU data, processing
                 is aborted.

                 This error should not occur regardless of
                 the DISKACCT setting in DEXOPS.  The message
                 is issued as a safety check during DEXSUS
                 processing to ensure that journal data is
                 written either to DETAIL or DAYS.

   ACTION:       Check to make sure that DEXPGEN has been run
                 successfully, and if so, contact CA MICS
                 Product Support.

   REFERENCES:   Section 7.3.1
```

```
+---------+
| DEX0098 |
+---------+

   TEXT:         >>>>%PROCESS HAS COMPLETED

   TYPE:         Information

   REASON:       Indicate that process has run to completion.

   ACTION:       None

   REFERENCES:   None


+---------+
| DEX0099 |
+---------+

   TEXT:         %LINE

   TYPE:         Information

   REASON:       Display a line of parameter input.

   ACTION:       None

   REFERENCES:   None
```

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

APPENDIX B

DATA DICTIONARY

```
 CCCCCCCC       AAAAAAA              MMM       MMM  IIIIIIIIII   CCCCCCCC    SSSSSSSS
CCCCCCCCCCC    AAAAAAAAA             MM MM   MM MM      III     CCCCCCCCCC  SSSSSSSSSS
CCC           AA       AA            MM  MM MM  MM      III     CCC         SS      SS
CCC           AA       AA            MM   MMM   MM      III     CCC         SS
CCC           AAAAAAAAAAA            MM    M    MM      III     CCC          SSSSSSSSS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC                 SS
CCC           AA       AA            MM         MM      III     CCC          SS     SS
CCCCCCCCCCC   AA       AA            MM         MM      III     CCCCCCCCCC   SSSSSSSSS
 CCCCCCCC     AA       AA            MM         MM  IIIIIIIIII   CCCCCCCC     SSSSSSS
```

r14.2

C A M I C S A N A L Y Z E R O P T I O N F O R V A X / V M S G U I D E

APPENDIX B

DATA DICTIONARY

APPENDIX B. DATA DICTIONARY

This appendix identifies each file in the CA MICS VAX/VMS Analyzer. This appendix shows the data sequencing, patterns of summarization, and a list of the data elements that appear on each file.

More information on the files in CA MICS VAX/VMS Analyzer can be found in the following sections.

1 - Common Data Elements
2 - VAX/VMS Accounting Information (DEA) Area
3 - VAX/VMS Monitor Information (DEM) Area
4 - VAX/VMS System Usage Information (DES) Area

B.1 - Common Data Elements

The following sections describe the common data elements used in the CA MICS VAX/VMS Analyzer:

1 - CA MICS Common Elements
2 - DEX Common Elements

B.1.1 - CA MICS Common Elements

The following common elements are used in the DEA, DEM, and DES information areas.

D A Y	Day of Month

The calendar day of the month, a number from 1 to 31.

TYPE: Common, retained - Numeric, measurement unit is integer.

SEE ALSO: ENDTS - End Time Stamp

ELEMENT DERIVATION: DAY is derived from the standard CA MICS time-stamp field ENDTS.

If the 13MONTHYEAR option of MICS.PARMS(SITE) is used to define your fiscal year as having 13 months of 28 days each, then the value of DAY is set according to how many days from the beginning of one of the thirteen "months" the day represented by ENDTS falls.

For example, if your fiscal year is specified as starting on July 1, then July 28 is DAY=28 of MONTH=01, while July 29 is DAY=01 of MONTH=02. The thirteenth "month" will have 29 days in a non-leap year and 30 days in a leap year.

RANGE OF VALUE: DAY takes any numeric value in the following ranges:

1 to 31

1 to 30 if the 13MONTHYEAR option is in use

SUMMARIZATION PROCESS: DAY contains the value of the last DAY encountered. It is not meaningful above the DAYS timespan.

USAGE NOTES: DAY can be used to select data for a day in which you are interested. Remember that files in the DETAIL and DAYS timespans may contain data for more than one day, and therefore selection is needed.

D A Y N A M E	Name of Day of Week

DAYNAME is the name of the day of the week associated with the measurement data in a CA MICS record.

TYPE: Common, retained, - Character, length 3.

SEE ALSO: DAY - Day of Month
ENDTS - End Time Stamp

ELEMENT DERIVATION: DAYNAME is derived from the standard CA MICS time-stamp field ENDTS.

RANGE OF VALUE: DAYNAME can contain any of the following:

SUN, MON, TUE, WED, THU, FRI, or SAT.

SUMMARIZATION PROCESS: DAYNAME contains the value of the last DAYNAME encountered. DAYNAME is not meaningful above the DAYS level.

USAGE NOTES: You can use DAYNAME to select data applicable to a day of the week in which you are interested. Remember that a cycle in the DETAIL or DAYS timespan may contain data from more than one day, so selection is necessary.

D U R A T I O N	Recording Interval Time

DURATION contains the number of seconds in the "reporting interval" of a given monitor. Reporting interval is the period of time during which a monitor gathers and reports statistics.

DURATION is calculated for the monitors RMF, TSO/MON, CICS, IMS, NPM, NetView, CA NetSpy, CA NetMaster, DB2, VM, VCC/VCA, TAP, and MQSeries. In these monitors, DURATION is the length of time that one record covers in the DETAIL timespan. In higher timespans, DURATION is the length of time during which the data collector provides measurement data within the interval that is covered by the CA MICS record for the time period.

The length of the RMF reporting interval is specified via the RMF INTERVAL parameter. Allowed values are from 1 to 60 minutes. The default value for the RMF data recording interval is 30 minutes for the RMF record types 70-79. The Hardware Instrumentation Services (HIS) create SMF type 113 records, also processed by the CA MICS Hardware and SCP Analyzer Option. The duration for SMF type 113 interval

records is 15 minutes.

The length of the TSO/MON reporting interval is specified via the INTVL parameter of the OPTIONS macro. Five minutes (INTVL=5) is the recommended value.

The length of a CICS recording interval is specified for CMF (CICS Monitoring Facility) at CICS/MVS Version 2 via the FREQ parameter in DFHMCT TYPE=RECORD macro. This affects the duration of the performance class global records. For CMF input at CICS/ESA Release 3.1.1 and higher, DURATION is set to the recording interval of CICS statistics data. You can change the statistics recording interval in CICS using the CEMT SET STATISTICS command. For TMON/CICS, the system interval records are written on a minute basis.

For IMS, you specify the number of IMS log records to record between IMS checkpoints. After these records are written to the IMS log, an IMS checkpoint is taken. Recognition of this checkpoint is the event CA MICS uses to produce another DETAIL IMS system record. You specify the number of log records between checkpoints with the CPLOG= parameter of the IMSCTF macro in the IMS Stage I generation. Note that under this scheme for logging checkpoints to the IMS log, DURATION is not a constant value for all CA MICS IMS DETAIL observations, and may, in fact, vary greatly according to IMS system load and time of day.

NPM, the NetView Performance Monitor, is divided into two subsystems: the Network Subsystem and the Session Subsystem. For the Network Subsystem, the length of the NPM recording interval is specified via the INTERVAL operand of the NPM INTERVAL initialization parameter and the online START COLLECT command. For the NPM Session Subsystem, the length of the interval is specified via the SESSINT operand of the NPM INTERVAL initialization parameter.

For NetView Session Monitor/NLDM, interval recording is provided through a Network Communications Control Facility (NCCF) CLIST. The CLIST issues an NCCF EVERY command to collect NLDM/RTM data on the interval you choose.

TCP information area files:

For all the files in the TCP area, DURATION is the elapse time between a connection start time and end time.

- o For TCP Termination (subtype 2) records, DURATION is calculated by subtracting connection start time from connection end time.

- o For TN3270E Telnet server SNA session termination (subtype 21) records, DURATION comes from field SMF119TN_NTDur at offset x'64' in SMF type 119.

- o For TSO Telnet client termination (subtype 23) records, DURATION comes from field SMF119TN_CTDur at offset x'4C' in SMF type 119.

- o For FTP client transfer completion (subtype 3) records, DURATION comes from field SMF119FT_FCDur at offset x'74' in SMF type 119.

- o For FTP server transfer completion (subtype 70) records, DURATION comes from field SMF119FT_FSDur at offset x'70' in SMF type 119.

- o For TCPIP Statistics (subtype 5) records that generate TCPIPX, TCPTCI, TCPUDI, and TCPICX files, the DURATION in these files comes from the respective fields SMF119AP_TSIPDuration, SMF119AP_TSTCDuration, SMF119AP_TSUDDuration, and SMF119AP_TSICDuration at offset x'00' in SMF type 119.

- o For Interface statistics (subtype 6) records, DURATION comes from field SMF119IS_IFDuration at offset x'00' in SMF type 119.

CDZ information area files:

For the CCT IBM Sterling Connect:Direct file, DURATION is th elapse time between time the COPY step started (STARTS) and ENDTS.

In DB2, the DURATION element for the DB2DSY and DB2DSD files represents the length of time that is reported in each detail observation. DB2 Release 2 provided DB2 systems personnel with the ability to define, at DB2 system startup, the time interval in which SMF system statistics are produced. This value is specified in minutes and should equal DURATION.

For VCA, except for the exceptions described below, DURATION contains the difference in time between the previous and current executions of the CA MICS Space Collector Option (VCC) scanning process.

For the first execution of a VCC scan, no previous scan timestamp is available. In this case, DURATION is set to the value on the OPTIONS statement specified in prefix.MICS.PARMS(VCAOPS), default 24 hours.

For data sets that do not have a data set creation time, whose create date is later that the previous VCC scan date, DURATION is calculated by multiplying the number of days the

data set existed prior to the current VCC scan, plus the time of day when the current VCC scan executed.

Beginning with z/OS 1.11, the Volume Table of Contents (VTOC) for Extended Address Volumes (EAV) began recording both the date and time of data set creation. For data sets with both the time and date of creation, if they were created after the previous VCC scan, DURATION is calculated as the difference between the current VCC scan time and data set creation time.

For data sets that are created after the VCC scan started, the DURATION value is set to zero.

For VM/370 Monitor, the recording interval length is specified via the INTERVAL operand of the MONITOR command. For VM/XA or VM/ESA Monitor, the interval length is specified via the INTERVAL operand of the MONITOR SAMPLE command.

In MQSeries, the DURATION element for the MQSBMS, MQSDMS, MQSLMS, MQSMMS, MQSCMS, and MQSDBS files represents the length of time that is reported in each detail observation.

The Message Queue Manager (MQM) allows an interval length specification at MQM system startup via the STATIME parameter. If not specified, the system default is 30 minutes. In the MQ Message Broker MQBMFA and MQBMFN files, the DURATION represents the difference between the ENDTS and STARTS of the message flow interval.

In all files for the CA MICS Analyzer Option for CA IDMS, except the IDMS System Activity File (IDMSSY), DURATION represents the elapsed time of the individual transaction records. In summarized timespans, DURATION represents the sum of DURATION of the individual transaction records that contributed to the summarized observation. DURATION is calculated as the difference between the End Date and Time (TASEDATE and TASETIME) and the Start Date and Time (TASSDATE and TASSTIME) fields from the CA IDMS Performance Monitor header record.

In all files for the CA MICS Analyzer Option for CA IDMS except the IDMS System Activity File (IDMSSY), the Start Time Stamp (STARTTS), and End Time Stamp (ENDTS), data elements reflect the start and end times of the individual transactions. For IDMSSY file, STARTTS and ENDTS are adjusted to represent the start and end of the time interval within which the transactions executed.

TYPE: Common accumulated data element - NUMERIC, measurement unit is INTEGER.

SEE ALSO: ENDTS - End Timestamp
STARTTS - Start Timestamp
INTERVLS - Number of Recording Intervals
SAMPLES - Number of Samples

ELEMENT DERIVATION: For RMF record types 70-79, DURATION contains the value input from field SMFxxINT. It is computed according to the following formula:

```
DURATION=60*FLOOR(SMFxxINT/100000)+MOD(SMFxxINT,100000)
/1000;
```

where SMFxxINT is in the form "MMSSTTT"

For the RMF type 113 record, DURATION is computed by subtracting STARTTS from ENDTS, input from the SMF type 113 record fields SMF113IntervalStart and SMF113IntervalEnd, respectively.

For TSO/MON, DURATION contains the value input from field TSMSINTV of the System Record (usually assigned SMF number 200).

For CICS, DURATION is measured from interval start to end based on timestamps in the record: DURATION=ENDTS-STARTTS. However, for CMF processing at CICS/ESA Release 3.1.1 and higher, DURATION is set to the sum of dispatch time (CSYQRDSP) and wait time (CSYQRWTM) when the statistics record is written for either the first or last (EOD) recording interval. The derivation is: DURATION=SUM(CSYQRDSP,CSYQRWTM).

For IMS, DURATION is calculated as described below unless the system interval time specification (from the IMSOPS member of the prefix.MICS.PARMS library on the OPTIONS statement) has been exceeded by 50 percent and the IMS checkpoint number of the previous checkpoint is not equal to the current checkpoint number-1. If these conditions are met, DURATION is set to the IMSOPS interval specification.

Under normal circumstances, the IMS DURATION contains the difference between the end timestamp of the current checkpoint and the end timestamp of the previous checkpoint. The interval specification on the IMSOPS OPTIONS statement does not override this calculation if the IMS checkpoint numbers monotonically increase (for example, 1, 2, 3, ...).

For example, assume that the specified system interval is 15 minutes. If the initial calculation of DURATION was 17 minutes, DURATION would contain a value of 17 minutes. If the initial calculation was 22.6 minutes (greater than 1.5 times 15 minutes, or 22.5 minutes), DURATION would be set to 15 minutes, except for IMS systems with monotonically

increasing checkpoint numbers.

For the NPM Network Subsystem, DURATION contains the value input from field LOGTIME, which is the actual NPA observation time for the NCP being used. For the NPM Session Subsystem, DURATION contains the difference between the interval end timestamp, VPVEINT, and the interval start timestamp, VPVDT.

For NLDM, DURATION contains the difference between the Collection Period Begin Time Stamp, LRTMCOLB, and the Collection Period End Time Stamp, LRTMCOLE.

For CA NetMaster, DURATION equals the elapsed time between session start and session end time.

For DB2 interval records, DURATION contains the difference between the end timestamp of the current SMF record and the end timestamp of the previously processed SMF record.

For VCC and VCA, DURATION is the difference between the time of the previous VCC scan execution, which is saved in the VCC control file, and the time of the current VCC scan execution. Exceptions to this computation, described earlier, occur when processing records with an initial VCC scan, and newly created data sets.

For the TLM component of the CA MICS Tape Analyzer Option, DURATION is the difference between the time of the previous TMCC scan execution, which is saved in the TMCC control file, and the time of the current TMCC scan execution.

For the VTS component of the CA MICS Tape Analyzer Option:

For the following SVT StorageTek VSM Virtual Tape Subsystem Information Area (SVT) DETAIL timespan files, DURATION is set to 900 seconds (15 minutes):

- o VTSS Subsystem Performance (SVTSSP) file
- o Channel Interface Performance (SVTCIP) file
- o Real Tape Drive Performance (SVTRPR) file
- o StorageTek VTSS Summary (SVTSVS) file

For the IBM Virtual Tape Subsystem Information (IVT) Area DETAIL and DAYS timespan files, DURATION is set to exactly 3600 seconds (1 hour).

For the IBM Hydra Virtual Tape Server Information (HVT) Area, DURATION contains the value input from field "Interval Duration" in the header section of each input record.

For the VM component, VMCCMU obtains DURATION from type 01 and type C1 account records columns 29-32. VMCDAC gets DURATION from types 02, C2, 03, and C3 account records columns 29-32. In VMCVMU file, DURATION contains the difference between VMUENDTS and STARTTS. In other VMC and VMX Information Area files, DURATION contains the difference between ENDTS and STARTTS in the DETAIL timespan.

For the CA MICS Analyzer Option for MeasureWare, DURATION is set to the value of the MeasureWare Global metric INTERVAL.

For the CA MICS Analyzer Option for MQSeries, DURATION contains the difference between the end timestamp of the current SMF record and the end timestamp of the previously processed SMF record. In the MQ Message Broker MQBMFA and MQBMFN files, the DURATION represents the difference between the ENDTS and STARTS of the message flow interval - the interval end datetime and interval start datetime.

For the CA MICS Analyzer Option for CA IDMS:

For the CA IDMS files, DURATION contains the difference between the ENDTS and STARTTS data elements. STARTTS and ENDTS contain the values of the start date and time (TASSDATE and TASSTIME) and end date and time (TASEDATE and TASETIME) fields from the CA IDMS Performance Monitor header record.

In individual IDMSUA file transaction records, DURATION is recorded in one of six category response time data elements (conversational, long, medium, short, excessive, and user-defined), based on the transaction type. Transaction type is determined in user written code in the relative longevity exit IDMRLRT.

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DURATION contains the sum of all occurrences of DURATION that were included in the level of summarization.

USAGE NOTES: At the DETAIL level, DURATION is no larger than ENDTS-STARTTS, but may be smaller if the data collector being used does not provide measurement data for all of the time between STARTTS and ENDTS.

For VCA, DURATION contains the time interval between VCC scan executions. However, if VCC processing is performed frequently, or if two or more devices are processed, the computed values may not provide the results you expect for computed average values on reports. It is difficult to

report average space utilization on volumes if the volumes are not processed with each VCC scan execution.

E N D T S	End Timestamp

ENDTS has different definitions, depending on the timespan and the data collector.

DETAIL Timespan:

- o ENDTS is the time of termination of an executable function, such as job, command, and so on.

- o For incident files in which ENDTS exists, ENDTS is the time at which the incident occurred:

  For the CICS, IMS, VM/CMS, and Network Analyzer Option incident files, this is the end timestamp of the transaction or system measurement interval during which the incident occurred.

- o For other files in which ENDTS exists, it is the time that the data collector's measurement interval ended:

  For CA IDMS data:

  For the IDMS System Activity File (IDMSSY), ENDTS is adjusted based on a user-defined interval value. This value is coded in the INTERVAL statement of prefix.MICS.PARMS(IDMOPS) and determines the cut-off point for each timespan.

  For all other CA IDMS files, ENDTS is the date and time when the IDMS transaction completed. ENDTS contains the value of the end date and time fields (TASEDATE and TASETIME) from the #PMTAS section of the CA IDMS Performance Monitor record.

  For RMF-derived data:

  For files generated from the SMF type 70-79 RMF records, ENDTS is computed by adding the duration of the interval to the starting time for compatibility with RMF.

  For files generated from the SMF type 113 subtype 1 or 2 CPU Hardware capacity, reporting, and statistics record generated by the IBM CPU Measurement Facility (CPU MF) Hardware Instrumentation Services (HIS).

  ENDTS is set equal to field SMF113_2_CTM (subtype 2) or field SMF113IntervalEnd (subtype 1), adjusted to local time by adding the GMT Offset value.

  For TSO/MON data:

  For the TSOTSO file, ENDTS is the time (date and time) when the TSO/MON record writer routine processed the record.

  For the TSOTSI and TSOTSA files, ENDTS is the time when the TSO command or subcommand ended.

  For the TSOTSU file, CA MICS combines user information from multiple TSO/MON system records when the user was inactive for one or more recording intervals. In this case, ENDTS is the timestamp from the last record consolidated into the DETAIL-level record.

  For the TSOTSC file, data is summarized by HOUR at the DETAIL level. (This is unusual for CA MICS, but is required by the volume of data.) ENDTS is taken from the timestamp of the last TSO/MON system record included in the hour.

  For VCC/VCA data:

  ENDTS is the time from the last execution of the VCC scan. If multiple VCC executions are processed into one update execution, then the value is the last time within that particular level of summarization.

  For the TLM component of the CA MICS Tape Analyzer Option:

  ENDTS is the time from the last execution of the TMCC scan. If multiple TMCC executions are processed into one update execution, then the value is the last time within that particular level of summarization.

  For the VTS component of the CA MICS Tape Analyzer Option:

  For the IBM Virtual Tape Subsystem Information Area (IVT) files, ENDTS is derived as follows:

  First, the date and time the SMF type 94 record was written to the SMF buffer is determined. The raw metrics in the record relate to the previous hour, so the code logic establishes ENDTS at 59 minutes and 59.99 seconds of that previous hour.

For example, if an SMF type 94 record was written at 13:12:48.02, the ENDTS for all IVTfff files generated from that record would be 12:59:59.99.

For the Sun StorageTek Information Area (SVT) files, ENDTS is derived as follows:

ENDTS is the time when the Sun StorageTek user SMF record was created. It is read from bytes 3 to 10 on the SMF record.

For VM Monitor interval recording data:

ENDTS in the CA MICS Analyzer Option for VM/CMS is the time when VM Monitor created the record.

For HP MeasureWare for HP-UX Systems data:

ENDTS in the CA MICS Analyzer Option for MeasureWare is the time that the data was captured in the log file.

For WebSphere MQ data:

In all files except the MQBMFA file, ENDTS is set to the value in the SMF header that tells the time and date the record was written to the SMF buffer, for example SM116DTE and SM116TME.

In the MQBMFA file, ENDTS is set to the interval end date and time values found in the message flow accounting record fields IMFLENDT and IMFLENTM.

For Web Analyzer:

ENDTS is set to the value input from field SM1209CQ of the SMF type 120 record subtype 9 at offset 68 (x'44') of the z/OS Request Information section.

All Other Timespans:

ENDTS is set to the maximum-end timestamp encountered from the records used to compile the summarized record. Thus, the combination of STARTTS and ENDTS in the observation identify the boundaries of the range of time over which the data in the record was summarized.

In the CA MICS Accounting and Chargeback Option journal files, ENDTS corresponds to the ENDTS in the component record being charged. In the ledger files, it is summarized from the journal file records.

TYPE: Common, retained - Numeric, measurement unit is SAS datetime value.

SEE ALSO: STARTTS - Start Timestamp
DURATION - Recording Interval Time
INTERVLS - Number of Recording Intervals

ELEMENT DERIVATION:

For RMF records, CA MICS follows the RMF convention and calculates ENDTS from the "beginning of interval" datetime given in fields SMFnnDAT and SMFnnIST, and the length of the interval, SMFnnINT.

For non-RMF records, ENDTS contains the value input from fields SMFnnDTE and SMFnnTME, where "nn" is the number of the SMF record. If the SMF record number is greater than 99, the input fields are named SMnnnDTE and SMnnnTME.

In the DAYS level of the ADMEXC file, ENDTS is set to the ENDTS of the CA MICS record from which the exception condition was detected.

For the CICS files, ENDTS is set to the following:

CICS CMF: The source of this data element is raw CICS monitor input data.

The CMF timestamp fields are presented in "TOD Clock Format," and are adjusted by the GMT offset for the system being processed. This offset is specified by the user in the CICOPS member of the MICS.PARMS library, on the CICOPTS statement. The field type is T, indicating a timestamp, an 8 byte copy of the output of a local store clock (STCK) instruction.

For transaction-level records written to the User Activity file (CICCSU) Application Unit Activity file (CICCAU) and the User Application Count file (CICCAC), ENDTS is either the time at which the user-task was detached or the time at which data recording was completed in support of CONV=YES. The source of this data element for these files follows:

Monitor	Input Field	Description
CTS 2.2 to 3.1	MNR_ID_STOP	Transaction stop time
CTS 3.2 and above	STOP (FID 006)	Finish time of measurement interval in the DFHCICS group

For system-level records written to the System Activity (CICCSY) file and the Dictionary file (CICCDC), ENDTS is the time at which the recording interval ended. The source of this data element for these files follows:

Monitor	Input Field	Description
CTS 2.2 and above	SMFSTDAT and SMFSTCLT	Collection date Collection time

For records written to the Incident file (CICCIN), ENDTS is the end time of the exception's occurrence. The source of this data element for this file follows:

Monitor	Input Field	Description
CTS 2.2 and above	EXCMNSTO	Exception stop time

ASG-TMON for CICS TS (TCE): The source of this data element is raw Monitor input data.

For transaction-level records written to the User Activity file (CICCSU) Application Unit Activity file (CICCAU), the User Application Count file (CICCAC), the File/DBD Activity file (CICCSF), and the CICS MRO Activity file (CICCMR), ENDTS is the time of either the transaction's termination or the conversational task's termination. The source of this data element for these files follows:

Monitor	Input Field	Description
TCE	TAENDDTS	Transaction end date

For system-level records written to the System Activity (CICCSY) file, ENDTS is the time at which the recording interval ended. The source of this data element for this file follows:

Monitor	Input Field	Description
TCE	TIENDDTS	Interval end date

For records written to the Incident file (CICCIN), ENDTS is the last available end timestamp from the Transaction (TA) records.

For The Monitor releases 7.1 and below, ENDTS is derived from the start timestamp of the next system record since the end timestamp is not in the raw data record.

CICS Transaction Gateway: For the CTG information area files created from the SMF type 111 CICS Transaction Gateway records, ENDTS is derived from the date and time values when the raw data record was written to the SMF buffer. The source of this data element follows:

Monitor	Input Field	Description
CTG	SMF111_SMFSTDTE SMF111_SMFTME	Date and time when record was moved to SMF buffer

For the DB2 files, ENDTS is set to the following:

ENDTS is derived from the date-time value found in the SMF header segment of the interval or thread event generated SMF record used to create the CA MICS database record.

For database records created from the interval based statistics records (SMF type 100, subtype 0 and 1), ENDTS correlates with the startup of the DB2 region and the user specified interval duration value (usually 30 minutes, but less if the region is stopped).

For database transaction-level records created from the SMF type 101, subtype 0 and 1 records, ENDTS correlates to an event, such as a thread termination, an authorization ID change in a thread re-use environment, or reaching the threshold value associated with DB2 ACCUMAC SMF record summarization.

For the IMS files, ENDTS is set to the following:

IMSISY (IMS System Activity) File: ENDTS contains the values input from the date and time fields found in the configuration checkpoint end record (type '40'X, subtype '98'X) on the IMS log.

IMSINC (IMS Incident) File: ENDTS contains the value input from the last ENDTS constructed for any other file.

Transaction-oriented Files (IMSISU User Activity, IMSIAU Application Unit Activity, and IMSIAC User Application Count Files): ENDTS depends on the collection method. With IMF, ENDTS is constructed from the end date and time presented by IMF in its DETAIL transaction accounting record (IMS log record type 'FA'X).

Without IMF, ENDTS is constructed from the date and time fields of the IMS log record that signifies the end of processing region occupancy for the message being analyzed. These fields may come from the type '31'X log record, field QLGUDTME, or the type '07'X log record, fields DLRDTE and DLRTME.

Except for those CA MICS Network Analyzer Option files listed below, ENDTS contains the value input from the SMF record end timestamp from the CA MICS Network Analyzer Option data sources. The exceptions, by file, are as follows:

- o NVSRTM and SNTNSV files - ENDTS is taken from the field LRTMCOLE of the RTM data section of the SMF type 39 record written by NetView/NLDM or CA NetMaster.

- o NVSNSA file - ENDTS is taken from field LACCENDT of the Accounting and Availability data section of the SMF type 39 record written by NetView/NLDM or CA NetMaster.

- o NVSNSC and NVSRTE files - There is only one CA MICS observation for each unique SNA session and the ENDTS for that observation is taken from the highest timestamp found in all SMF type 39 records (written by NetView/NLDM or CA NetMaster) for that session.

- o CDZCCT file - ENDTS is derived from the date-time value found in the SMF header segment of the IBM Sterling Connect:Direct copy termination record, SMF 133 (user defined), subtype CT.

For all VSE/Power files, the ENDTS is derived from the ACDATE and ACSTOP fields of the account record used to create the observation.

For the VMC and VMX files, the derivation of ENDTS depends on various sources:

- o VM and VM account data - VMCCMU, VMCDAC, VMCNDT and VMCVRA files obtain ENDTS from record types 01, C1, 02, C2, 03, C3, 04, 05, 06, 08, C0, 0C (RSCS) records columns 23-28. VMCNTU obtains ENDTS from record type 07 columns 53-60.

- o VM/ESA and z/VM monitor data - For files that have one observation per input record, the value of ENDTS is taken directly from the time-of-day clock stamp (MRHDRTOD) in the record header of input record. For other files, the value of ENDTS is taken from the record with the highest timestamp that contributes to the observation.

For CA MICS Analyzer Option for MeasureWare data, ENDTS is set to the value of the configuration metric DATE_SECONDS.

RANGE OF VALUE: ENDTS contains any valid SAS datetime value. The value of this data element is not affected by use of the 13MONTHYEAR option in prefix.MICS.PARMS(SITE).

SUMMARIZATION PROCESS: ENDTS contains the value of the last occurrence of ENDTS in the observations processed for the summarization level.

USAGE NOTES: The value of DURATION, which is the amount of time in the measurement interval, is sometimes smaller than ENDTS minus STARTTS. This occurs when measurement data was not available from a data collector for some of the time in the interval. A common mistake is to compute averages using ENDTS minus STARTTS.

H O U R	HOUR OF DAY

The hour of the day, based on a 24 hour clock.

TYPE: Common, retained - Numeric, measurement unit is integer.

SEE ALSO: ENDTS - End Time Stamp

ELEMENT DERIVATION: HOUR is derived from the standard CA MICS time-stamp field ENDTS.

RANGE OF VALUE: HOUR takes any numeric value in the following range:

00 through 23, where 00 represents the hour 00:00:00.00 through 00:59:59.99 and 23 represents the hour 23:00.00.00 through 23:59:59.99.

SUMMARIZATION PROCESS: HOUR contains the value of the last HOUR encountered. HOUR is not meaningful above the WEEKS timespan.

At the DAYS and WEEKS levels of summarization in most CA MICS files, HOUR is used to determine a unique key for records within a file based on the timespan so other elements in the file can be summarized.

USAGE NOTES: You can use HOUR to select data for a reporting period.

```
+-----------------+
| I N T E R V L S |   Number of Recording Intervals
+-----------------+
INTERVLS is the number of measurement intervals that occurred
over a specific measurement period (DURATION).  It is the
number of monitor logical records.

A data collector collects its measurements for a user-defined
period of time and then writes them to an output record.
Each such period of time is termed an "interval" and its
length, which extends from the record's starting timestamp
(STARTTS) to its ending timestamp (ENDTS), is termed its
duration.  The derivation is potentially unique to each of
the data collectors.

TYPE:  Common retained data element --  NUMERIC, measurement
unit is INTEGER.

SEE ALSO:  DURATION - Recording Interval Time
           STARTTS  - Start Time Stamp
           ENDTS    - End Time Stamp

ELEMENT DERIVATION:  For RMF, CICS, IMS, Network, TAP, and
MQSeries analyzer files in the DETAIL timespan, INTERVLS=1.

For the TLM component of the CA MICS Tape Analyzer Option,
INTERVLS is always one at the detail timespan.

For TSO/MON files in the DETAIL timespan, INTERVLS only
exists in the TSOTSO, TSOTSU, and TSOTSC files, which are all
derived from the TSO/MON Systems Record.

INTERVLS=1 for all records in the TSOTSO file in the DETAIL
timespan.

INTERVLS may be 1 or greater in the TSOTSU file.  CA MICS
combines user information from multiple TSO/MON Systems
Records when the user was completely inactive for one or more
TSO/MON recording intervals.  In this case, INTERVLS is equal
to the number of combined recording intervals.

Since the TSOTSC file is summarized to the hour in the DETAIL
timespan (necessary for data compression), INTERVLS is equal
to the number of recording intervals that occurred during
the hour (usually 12 at the recommended TSO/MON Systems
Record recording interval of 5 minutes).

For CA MICS Analyzer Option for VM/CMS files in the DETAIL
timespan:

In VMCVMU and all VMX interval recording files, INTERVLS is
computed by dividing DURATION of a record by the length of
recording interval.

In VMX event files and other VMC files in which INTERVLS
exists, INTERVLS is set to 1.

RANGE OF VALUE:  The value of this element ranges from a
minimum of 0 to a maximum that is limited only by practical
bounds.

SUMMARIZATION PROCESS:  INTERVLS contains the sum of all
occurrences of INTERVLS that were included in the level of
summarization.


+-----------------+
| M I C S V E R   |   CA MICS VERSION NUMBER
+-----------------+
The CA MICS version number identifies the release level of
the CA MICS platform.

TYPE:  Not applicable.

SEE ALSO:  Not applicable.

ELEMENT DERIVATION:  The MICSVER data element is no longer
needed in any of the CA MICS files because it has no further
use in the CA MICS product line.  Consequently, it has been
obsoleted.  However, it remains in several historical files
and so this definition member has been preserved.

RANGE OF VALUE:  MICSVER contains any value in the following
ranges:

  '01' through '99' and 'AA' through 'ZZ'.

SUMMARIZATION PROCESS:  Not applicable.

USAGE NOTES:  None


+-----------------+
| M O N T H       |   Month of Year
+-----------------+
MONTH is the calendar month of the year (1 through 12).

TYPE:  Common, retained - Numeric, measurement unit is
integer.

SEE ALSO:  DAY - Day of Month

ELEMENT DERIVATION:  MONTH is derived from the standard
CA MICS time-stamp field ENDTS.  The value ranges from 1
```

through 12, representing January through December.

If the 13MONTHYEAR option is specified in MICS.PARMS(SITE), then the value of MONTH ranges from 1 to 13, representing the thirteen months in the organization's fiscal year. In this case, the fiscal year is divided into groups of 28 days, and MONTH is set according to which group of 28 days the date/time falls. For example, if the fiscal year was specified as starting on July 1, July 28 would be the last day of MONTH=01 and July 29 would be the first day of MONTH=02.

The MONTH variable in the CA MICS Accounting and Chargeback financial recap files, ACTRCP and ACT_RC, is set to the month value of the prior month. MONTH in these files represents the month for which the work was invoiced, not the month in which the work occurred.

RANGE OF VALUE: MONTH takes any value in the following ranges:

1 to 12; 1 to 13 with 13MONTHYEAR option.

SUMMARIZATION PROCESS: MONTH contains the last value of MONTH processed in the level of summarization.

At the higher levels of summarization in most CA MICS files, MONTH is not summarized. It is used to determine a unique key for records within a file based on the timespan, so other elements in the file can be summarized.

O R G S Y S I D — Original System Identification

For z/OS systems, ORGSYSID is the SMF system identifier of the z/OS system on which data was collected.

The z/OS SMFWTM routine copies field SMCASID to offset x'0E' of every SMF record before it writes it to a SYS1.MANx data set. SMCASID is filled in during the IPL process from the SID= parameter in member SMFPRMxx of SYS1.PARMLIB.

For non-z/OS systems, ORGSYSID is not obtained directly from the input data. It is obtained from specifications in prefix.MICS.PARMS member cccOPS.

TYPE: Common retained data element - CHARACTER, LENGTH 4.

SEE ALSO: SYSID - System Identification

ELEMENT DERIVATION: When z/OS SMF is the source of the input records, ORGSYSID is extracted from the field at offset 10 in the standard SMF header area. In IBM documentation, this field has the name SMFxxSID.

Although the VCA component normally reads SMF records written by the VCC component, it will also read non-SMF SMF-look-a-like records written by VCC to a sequential data set.

For all AUM, DEX, VSE/PWR, and SRL files, ORGSYSID is not obtained directly from the data. It is obtained from specifications in prefix.MICS.PARMS member cccOPS.

For CICS, when reading data written by ASG-TMON for CICS, the ORGSYSID is equated to the field called xxSMFSID in the ASG-TMON for CICS documentation. The 'xx' stands for the record type: TA, TI, or TR.

For IMS, when reading data extracted from the BMC MAINVIEW for IMS Online product and the IMS Analyzer AutoDetect Facility is being used, the ORGSYSID value is extracted from the MAINVIEW transaction accounting records, specifically from the field containing the System Name ID (SYSNAME). Otherwise, the SYSID specified on the IMSOPS OPTION statement's OSYSID parameter is assigned to all the incoming log data.

NOTE: A more correct approach to populating the ORGSYSID value using the IMS Analyzer AutoDetect Facility would be to use the field that contains the SMF ID to populate the ORGSYSID data element. However, the BMC Mainview transaction accounting records do not always contain the SMF ID. If your System Name IDs (SYSNAME) differ from the SMF IDs defined for the z/OS LPARs, the IMS00408 error message displays in the MICS log with information about the SYSNAME and SMF ID values so that you can make appropriate adjustments to the OSYSID parameter of the IMSOPS OPTION statement.

Tandem MEASURE data contains a Tandem system identifier. The CA MICS Tandem Option matches this Tandem system identifier to one of those specified in the TDMOPS parameter library member to identify the Tandem system and obtain the ORGSYSID.

For the TLM component of the CA MICS Tape Analyzer Option, ORGSYSID represents the tape library name (TAPELIB) used to identify unique tape library systems. ORGSYSID is derived from the TMCTLIB field of the TMCC record. TMCTLIB is the tape library name specified during TMCC execution. ORGSYSID is not used in the sort sequence of TLM files and is not retained outside the DETAIL timespan.

For the VTS component of the CA MICS Tape Analyzer Option, since no ORGSYSID is supplied by VTS data sources, ORGSYSID is not used. Data element SYSID is assigned the value SYS1 by default, unless overridden with a VTSSYSID statement in prefix.MICS.PARMS(VTSOPS).

VM measurement data does not contain the SMF System Identifier. The CA MICS Analyzer Option for VM/CMS matches the CPU model and serial number in the monitor data to an ORGSYSID definition in the VMCOPS parameter library member to identify a VM system and obtain the ORGSYSID.

For VSE/PWR files, this field is obtained from the VSESYSID based on the mapping defined in prefix.MICS.PARMS(PWROPS).

For the CA MICS Web Analyzer component processing Web Log Extractor (WLE) records, the ORGSYSID is supplied by configuration file server information parameters in prefix.MICS.PARMS(WLECONF).

For a non-MSI FDA component, ORGSYSID is populated by code in the USRFMT2 macro. The MSI FDA uses the ORGSYSID statement in its fdaGENIN member to identify the input field containing the ORGSYSID. The MSI FDA could also use the cccSYSID exit for specialized ORGSYSID assignment.

RANGE OF VALUE: 1 to 4 alphanumeric characters.

SUMMARIZATION PROCESS: ORGSYSID contains the value of the last ORGSYSID encountered.

USAGE NOTES: CA MICS provides a mapping of input SYSID (ORGSYSID) to logical SYSID (SYSID) in order to support the concept of "successor" systems.

The ORGSYSID is used to find information about the computer system that is identified by the ORGSYSID, information such as the CPU model and MIPS rating.

| S T A R T T S | Start Timestamp

STARTTS has different definitions depending on the timespan and the data collector.

DETAIL Timespan:

- o Gives the time of initiation for an executable function, such as job or command.
- o Tells when a measurement interval started.

All Other Timespans:

- o Gives the minimum start timestamp encountered in the records used to compile the summarized record.
- o In any timespan but DETAIL, the combination of STARTTS and ENDTS in the record identifies the boundaries of the range of time over which the data in the record was summarized.

In the CA MICS Accounting and Chargeback Option ledger files, STARTTS is the lowest ENDTS value in the journal files from which the ledger file was summarized.

TYPE: Common, retained - Numeric, measurement unit is SAS datetime value.

SEE ALSO: DURATION - Recording Interval Time
ENDTS - End Timestamp

ELEMENT DERIVATION: The derivation of STARTTS depends on the file:

RMF Files: For files generated from the SMF type 70-79 RMF records, STARTTS is derived from fields SMFnnDAT (interval start date) and SMFnnIST (interval start time) of each RMF record (where nn is the record number).

For files generated from the SMF type 113 subtype 1 or 2 Hardware capacity, reporting, and statistics record generated by the IBM CPU Measurement Facility (CPU MF) Hardware Instrumentation Services (HIS).

For 113-2 records, STARTTS is determined in one of two ways:

- If the interval record is the first from a HIS collection run, STARTTS is set equal to field SMF113_2_CTS, adjusted to local time by adding the GMT Offset value.
- If the interval record is an intermediate or final from a HIS collection run, STARTTS is set equal to the ENDTS of the previous interval record.

For 113-1 records, STARTTS is set equal to field SMF113IntervalStart, adjusted to local time by adding the GMT Offset value.

- If the interval record is the first from a HIS collection run, STARTTS is set equal to field SMF113_2_CTS, adjusted to local time by adding the GMT

Offset value.

- If the interval record is an intermediate or final from from a HIS collection run, STARTTS is set equal to the ENDTS of the previous interval record.

TSO Files: For the TSOTSO file, this value is calculated by subtracting the length of recording interval (TSMSINTV) from the record generation date (TSMSDTSP) and record generation time (TSMSTMSP) fields in the systems segment of the TSO/MON Systems Record.

For the TSOTSU and TSOTSC files, STARTTS is the same as either the TSOTSO file or the LOGON timestamp, TSMSLOGD (LOGON date) and TSMSLOGT (LOGON time), whichever is less.

For the TSOTSI file, STARTTS is taken from the Command Start Time (TSMPSTRT) and Command Start Date (TSMPSTRD) fields in the TSO/MON Command Record.

SMF Files: For the BATJOB file, STARTTS is taken from fields SMF30DTE and SMF30TME in the Common Address Space Work Record (type 30), subtype 5.

For the BATPGM, BAT_ST, BAT_TS, and BAT_TP files:

o For complete step records (PGMINTVL='SE24'), STARTTS is taken from fields SMF30STD (date initiator selected step) and SMF30SIT (time initiator selected step) of the SMF Common Address Space Work Record (type 30).

o For interval records, STARTTS represents the start time of the interval and is taken from one of the following, depending on the level of z/OS that produced the SMF record:

- The end time of the previous interval
- Fields SMF30IST and SMF30ISD
- Field SMF30ISS

For the BATSPL file, STARTTS is taken from fields SMF6WSD and SMF6WST (datetime output function began work on this file) of the SMF JES2 and JES3 Output Writer records (type 6).

ADM Files: For ADMEXC at the DAYS level, STARTTS is undefined. For ADMEXC at the MONTHS level, STARTTS is the lowest date and time associated with data summarized into the record.

CA IDMS Files: For the IDMS System Incident File (IDMSIN), STARTTS is the date and time when the CA IDMS transaction began processing. STARTTS contains the value of the Start Time (PMHSTIME) field from the header section of the CA IDMS Performance Monitor record.

For all other CA IDMS files, STARTTS is adjusted based on a user-defined interval value. This value is coded in the INTERVAL statement of prefix.MICS.PARMS(IDMOPS) and determines the cut-off point for each timespan.

CICS Files: For the CICS Incident File (CICCIN), STARTTS is set to the value of the ENDTS for the incident. The STARTTS value for the Application Unit Activity File (CICCAU) is set from and identical to the STARTTS value for the User Activity File (CICCSU). Construction of the timestamp itself varies according to the monitor used to collect CICS data:

CMF: When type 110 Performance records are input, the STARTTS is read from Field 5 (START) that carries either the time at which the user task was attached or the time at which data recording was most recently reset in support of the MCT user event monitoring point DELIVER option or of the monitoring options MNCONV, MNSYNC, or FREQUENCY. After this value is read from Field 5, it is adjusted by the GMT offset that is input from the header section of the same type 110 record. The resulting value populates the STARTTS in the CICS User Activity (CICCSU) file and in the CICS Application Unit Activity (CICCAU) file.

When type 110 Statistics records are input, the STARTTS is calculated by the formula:

ENDTS - DURATION

The STARTTS values in this case populate the CICS System Activity (CICCSY) file.

ASG-TMON for CICS TS: To populate the CICS User Activity (CICCSU) file, the STARTTS is read from the main segment of the type TA record. To populate the CICS System Activity (CICCSY) file, the STARTTS is read from the main segment of the type TI record.

DB2 Files: For the DB2DSU, DB2DSP, DB2DDU, and DB2DDP files, STARTTS is set equal to ENDTS minus the elapsed time, DSUELPTM. For the DB2DSD, DB2DSY, and DB2DDY files, STARTTS is set equal to the ENDTS of the previous record.

IMS Files: For the IMS Incident File (IMSINC), STARTTS is

equal to and set from the ENDTS of the incident. The STARTTS values for the Application Unit Activity File (IMSIAU) and User Application Count File (IMSIAC) are set from and identical to the STARTTS value for the User Activity File (IMSISU). For the System Activity File (IMSISY), STARTTS = ENDTS - DURATION. Construction of STARTTS for transaction-oriented files varies according to the method used to accumulate IMS data:

With IMF: STARTTS is constructed from the time and date of processing region occupancy presented by Control/IMS in its detail transaction record.

Without IMF: STARTTS is constructed from the time and date fields in the log type '31'X record for the Get Unique of the transaction's first message segment from the IMS message queue. The field in the '31'X record that contains date and time is QLGUDTME.

MQS Files: For all files except MQAQAA, MQATAA, and MQBMFA, the STARTTS for an interval is the ENDTS of the previous interval.

For the MQAQAA and MQATAA files, STARTTS is derived by subtracting Task Execution Time (TAAEXCTM) from End Timestamp (ENDTS): STARTTS = ENDTS - TAAEXCTM;

Note that TAAEXCTM = TAAIETS-TAAISTS where TAAIETS is SMF field WTASINTE and TAAISTS is SMF field WTASINTS in IBM DSECT CSQDWTAS.

For the MQBMFA file, STARTTS is set to the interval start date and time values found in the Message flow accounting record fields IMFLSTDT and IMFLSTTM.

For Web Analyzer:

STARTTS is set to the value input from field SM1209CM of the SMF type 120 record subtype 9 at offset 4 (x'04') of the z/OS Request Information section.

Network Analyzer Option Files:
NPM Network data collection: STARTTS is the end timestamp of the measurement interval minus the actual observation time of the interval: STARTTS = ENDTS - NPMTMAOT.

NPM Session data collection: STARTTS is the Date and Time of Beginning Interval, VPVDT, from the Session Statistics Detail Record.

NetView/NLDM or CA NetMaster NTS data (by CA MICS File):

- o NVSRTM and SNTNSV files - STARTTS is taken from field LRTMCOLB of the SMF type 39 record.
- o NVSNSA file - STARTTS is taken from field LACCBEGT of the Accounting and Availability data section of SMF type 39 record.
- o NVSNSC and NVSRTE files - There is only one observation in CA MICS for each unique SNA session in the DETAIL timespan of these files, and the STARTTS for that observation is taken from the lowest timestamp found in all SMF type 39 records processed for that session.

NetView/NPDA or CA NetMaster NEWS Data: STARTTS is set equal to ENDTS.

IBM Sterling Connect:Direct data collection: STARTTS is the timestamp for when the COPY function was started. It is a combination of fields CTSDATE and CTSTIME from the SMF 133 (user defined) record, subtype CT.

VSE/Power Files: The element is derived from the ACDATE and ACSTRT fields of the account record used to create the observation. However, if the value of ACSTRT is greater than ACSTOP, the date used is one day earlier than ACDATE.

When processing the Execution Account Record for VSE/Power, the element is derived by subtracting the execution time (EXCJDUR) from ENDTS.

VM/CMS Files: In the DETAIL timespan, the derivation of STARTTS varies on files:

For VMCCMU, VMCDAC, and VMCVRA, STARTTS contains the difference between ENDTS and DURATION. In VMCNTU file, STARTTS contains the value input from record type 07 column 45-52.

For VMXVXU, VMXVXT, VMX_VU, and VMX_VT, STARTTS of a user record is set to the value of ENDTS of the previous record processed for the user. For the first user record after a user logged on, STARTTS is set to either the logon timestamp, if available, or the difference between ENDTS and interval record length.

For other files, STARTTS contains the value of ENDTS of the previous record processed under the same unique key for records within a file.

TLM Component TMC Information Area files: For the TMCfff files, STARTTS is the start of the collection interval. Generally, this is the last time TMCC was executed for a given tape library. STARTTS is derived from the TMCLDATE/TMCLTIME fields of the TMCC record.

VTS Component IVT Information Area files: For the IVTfff files, created from SMF type 94 records for IBM VTS systems, STARTTS is established as the exact hour value of the hour that preceded the hour when the SMF type 94 record was written to the SMF buffer.

For example, if an SMF type 94 record was written at 13:12:48.02, the STARTTS for all IVTfff files generated from that record would be 12:00:00.00.

VTS Component SVT Information Area files: For the SVTSSP, SVTCIP, and SVTRPR files, STARTTS is created from Sun StorageTek User SMF records. It is calculated by subtracting 15 minutes from the ENDTS for the interval records, subtypes 10, 11, and 20.

For all other SVT files except SVTSVS, STARTTS is set to the ENDTS value read from the StorageTek event records.

For the SVTSVS file, STARTTS is set to the start of each 15 minute boundary 00:00.00, 15:00.00, 30:00.00, 45:00.00, in the DETAIL timespan.

RANGE OF VALUE: Any valid SAS datetime value. The value of this data element is not affected by your use of the 13MONTHYEAR option in prefix.MICS.PARMS(SITE).

SUMMARIZATION PROCESS: The minimum STARTTS encountered in summarizing a number of records to a higher level of granularity.

S Y S I D — SYSTEM IDENTIFICATION

SYSID is the translated system identifier that was derived from ORGSYSID by the CA MICS SYSID translation process. It is defined in the CPLXSID or SYSID member of a CA MICS parameter library.

TYPE: Common, retained - Character, length 4.

SEE ALSO: ORGSYSID - Original System Identifier

ELEMENT DERIVATION: Every record coming from most CA MICS data collection sources contains an SMF system identifier in the appropriate field of the record. CA MICS calls this field the ORGSYSID. For collection sources that do not provide such an identifier, CA MICS provides a method to determine the ORGSYSID. This ORGSYSID is compared to a user-defined SYSID table (PARMS members CPLXSID and/or SYSID) to determine if the ORGSYSID should be used as the SYSID or translated to a reporting SYSID.

The prefix.MICS.PARMS(SYSID) member is used to build SYSID table entries for a specific unit. The sharedprefix.MICS.PARMS(CPLXSID) member is used to build SYSID table entries for every unit in a complex.

RANGE OF VALUE: SYSID can contain any combination of 1 to 4 alphabetic and/or numeric characters.

SUMMARIZATION PROCESS: SYSID is not summarized. It is used to determine a unique key for records within a file based on the timespan, so other elements in the file can be summarized.

USAGE NOTES: Most CA MICS files have SYSID as their highest level or primary key. This is what allows CA MICS to store information from multiple systems (SYSIDs) in the same database.

The ADMIHL file uses ORGSYSID as its highest level or primary key.

The CA MICS Network Analyzer Option uses SYSID as its second highest level key. The highest level key in these files is NETWRKID, Network Identifier. NETWRKID enables these files to support data from multiple networks (for example, networks supported by SNA Network Interconnect, SNI).

For the VTS component of the CA MICS Tape Analyzer Option, in all IBM Virtual Tape Subsystem Information Area (IVTfff) files, SYSID defaults to SYS1, or may be set to a value specified in prefix.MICS.PARMS(VTSOPS) using the VTSSYSID statement.

For the TLM component of the CA MICS Tape Analyzer Option, SYSID is not used in the sort sequence of TLM files, and is not retained outside the DETAIL timespan.

There is no general way of meaningfully combining or even comparing data gathered on different computer systems, so be careful when putting together data from two SYSIDs.

```
+-----------------+
| W E E K         |   WEEK OF YEAR
+-----------------+
WEEK is the calendar week of the year (an integer from 1 to
54).  To CA MICS, a week extends from Sunday through Saturday
(unless changed via the WEEKSTART parameter mentioned below).
The first and last weeks of the year may have less than seven
days.

WEEK=54 can only occur in a leap year, and then only when
January 1 falls on a Saturday (and therefore is a week all by
itself).

TYPE:  Common, retained - Numeric, measurement unit is
integer.

SEE ALSO:  DAY     - Day of Month
           MONTH   - Month of Year

ELEMENT DERIVATION:  WEEK is calculated from the date portion
of the variable ENDTS.  WEEK=01 begins on January 1 and
extends through the following Saturday.  WEEK=02 begins on
Sunday, and so forth.

If you specified the 13MONTHYEAR option of MICS.PARMS(SITE),
then WEEK=01 begins on the first day of the fiscal year
specified and extends up to the end of the following
Saturday.  If you have modified the definition of a standard
week via the WEEKSTART parameter in MICS.PARMS(SITE), then
your WEEK=01 will begin on the first day of your fiscal year
and extend to the end of the day before that specified at the
operand of the WEEKSTART statement.

RANGE OF VALUE:  WEEK takes any numeric value in the range 1
through 54.

SUMMARIZATION PROCESS:  WEEK is set to the value of the last
WEEK encountered in the summarization period.

For files that exist in the WEEKS timespan, WEEK is not
summarized.  It is used to determine a unique key for records
in the WEEKS timespan so other elements in the file can be
summarized.


+-----------------------------+
| Y E A R   a n d   Y E A R 4 |   Year of Century
+-----------------------------+
The YEAR (the encoded year of the century) and YEAR4 (the
four digit year) data element descriptions are presented in
the following order:

* YEAR  -- Year of Century
* YEAR4 -- Year

+-----------------+
| Y E A R         |   Year of Century
+-----------------+
YEAR is the "cyy" encoded year of the century where yy is the
last two-digits of year and c, the century indicator, is 0
for years before 2000 and 1 beginning in year 2000.  For
example, the year 1998 is represented as 98 while the year
2001 is represented as 101.

You can also think of YEAR as the 4-digit year minus 1900.

YEAR is carried in encoded form to reduce database space
while preserving proper chronological sort sequence (e.g.,
data for year 2000 sorts after data from 1999).  Since YEAR
appears in every observation of every file, a savings of
several megabytes is possible.  The maximum value of the
encoded YEAR element is 255, which supports years through
2155 (1900 + 255).

TYPE:  Common, retained - Numeric, measurement unit is
integer.

SEE ALSO:  MONTH - Month of Year
           WEEK  - Week of Year

ELEMENT DERIVATION:  Normally, CA MICS calculates YEAR by
subtracting 1900 from the year part of ENDTS.

If 13MONTHYEAR YES was specified in prefix.MICS.PARMS(SITE),
YEAR is set to the year in which the fiscal year defined by
the time stamp begins minus 1900.  In other words, if your
site specifies that your fiscal year runs from July 1 of one
year to June 30 of the following year, then YEAR would be set
to 201 for any date/time from 00:00:00 on July 1, 2001, to
23:59:59 on June 30, 2002.

The YEAR variable in the CA MICS Accounting and Chargeback
financial recap file is set to the year value of the prior
period.  In all of the financial recap files:  ACTRCP,
ACT_RC, ACT_YR, ACT_YT, ACT_RR, AND ACT_RT, the variable YEAR
represents the year for which the work was invoiced.

RANGE OF VALUE:  YEAR takes any numeric value in the range 0
to 255.

SUMMARIZATION PROCESS:  YEAR contains the value of the last
YEAR encountered during the summarization process.

At the higher levels of summarization in most CA MICS files,
```

this is a summarization control variable. During the summarization process a new record is created for each unique record key combination, and this field is part of the key of its record.

USAGE NOTES: The default output format for YEAR, YR., prints the last 2-digits of year to retain the "year of century" presentation. For example, for data from year 2001, YEAR is printed as 01 rather than the actual value of 101.

For some applications, such as in program logic, it may be helpful to work with the full 4-digit year. The %YEAR4 macro is available for this purpose. For example, the following SAS instructions both select data for the years 1998 through 2001.

```
IF 98 LE YEAR LE 101;

IF 1998 LE %YEAR4 LE 2001;
```

Y E A R 4	Year

YEAR4 is the four digit year.

TYPE: Common, retained - Numeric, measurement unit is integer.

SEE ALSO: MONTH - Month of Year
WEEK - Week of Year
YEAR - Year of Century

ELEMENT DERIVATION: CA MICS calculates YEAR4 as the value of the element YEAR plus 1900.

RANGE OF VALUE: YEAR4 takes any numeric value in the range 0 to 2155.

SUMMARIZATION PROCESS: This value is not summarized.

USAGE NOTES: The YEAR4 data element is shipped turned off. Furthermore, it is not defined as a normal CA MICS data element in sharedprefix.MICS.GENLIB. As a result, you cannot use the standard method of file tailoring to turn YEAR4 on or off. You must activate or deactivate this data element via a parameter in sharedprefix.MICS.PARMS(CPLXDEF). Since CPLXDEF affects the operation of the entire CA MICS complex, the data element will either reside in every CA MICS file or none at all.

Refer to section 2.3.1.8 of the "PIOM" for information on activating/deactivating the YEAR4 data element.

Z O N E	Time Zone

ZONE is the time zone during which the measurements in this record were taken.

ZONE enables you to categorize activity at a level of granularity that is greater than HOUR but less than DAY. (This is in the DETAIL timespan; ZONE can contain much more time than a day in the higher timespans.)

Each installation defines ZONEs in the manner that is most meaningful to it, but most installations include the notion of "prime" vs. "non-prime" time in the ZONE definitions.

The definition and use of ZONEs is discussed in Section 2.3.2.3 of the CA MICS Planning, Installation, Operation, and Maintenance (PIOM) Guide.

TYPE: Common, retained - Character, length 1.

SEE ALSO: HOUR - Hour of Day

ELEMENT DERIVATION: CA MICS derives ZONE by comparing the date and time (ENDTS) of the measurement source record with your site's HOLIDAY specifications. If there is a match, the "holiday" ZONE is assigned. If there is not, CA MICS compares the DAY and HOUR with the your ZONE specifications in member ZONE of prefix.MICS.PARMS. A match will always be found because MICSGEN requires that all 168 hours within a week be assigned to a ZONE.

RANGE OF VALUE: ZONE contains any one of the following characters:

1, 2, 3, 4, 5, 6, 7, 8, or 9.

SUMMARIZATION PROCESS: ZONE is set to the value of the last ZONE encountered in the summarization process.

At the higher levels of summarization in most CA MICS files, ZONE is not summarized. It is used to determine a unique key for records within a file based on the timespan, so other elements in the file can be summarized.

USAGE NOTES: ZONE is not affected by the 13MONTHYEAR option in prefix.MICS.PARMS(SITE).

B.1.2 - DEX Common Elements

The following DEX common elements are used in the DEA, DEM, and DES information areas.

D E X A C C N T Account Name

DEXACCNT contains the account name associated with the process.

TYPE: Retained - Character, length 16.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXACCNT is set to the value of the string pointed to by the ACR$W_ACCOUNT field of the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: DEXACCNT can contain any combination of 1 to 16 alphanumeric characters, including the symbols '_' and '$'.

SUMMARIZATION PROCESS: DEXACCNT contains the last value of DEXACCNT processed in the level of summarization.

USAGE NOTES: DEXACCNT is a set up by the VAX system manager for a group of users to be treated similarly for resource accounting. The first account name associated with a group of users (based on the first 16 bits of the user identification code) is part of the owner of files and other resources created by the user.

D E X B O O T S Monitor System Boot Time Stamp

The date and time of the last boot of the system.

TYPE: Common data element - SAS datetime value.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXBOOTS is obtained from the VMS MONITOR SYSTEM INFORMATION record (type 129) from MNR_SYI$O_BOOTTIME.

RANGE OF VALUE: Any valid SAS time-stamp value. The value of this data element is not affected by your use of the 13MONTHYEAR option in prefix.MICS.PARMS(SITE).

SUMMARIZATION PROCESS: DEXBOOTS contains the value of the earliest occurrence of DEXBOOTS in the observations processed for the summarization level.

USAGE NOTES: None

D E X D E V I C Device Name

DEXDEVIC contains device designation from VMS, as displayed by SHOW DEVICE.

TYPE: Common, retained - Character, length 16.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXDEVIC is set to the value in the DEV field on the following separator records from the DEXSUS collect routine:

- DEXSUS DISK DEVICE separator record
- DEXSUS DISK QUOTA separator record
- DEXSUS DISK USAGE separator record

RANGE OF VALUE: DEXDEVIC can contain any combination of 1 to 16 alphanumeric characters, limited by the device naming conventions of VMS.

SUMMARIZATION PROCESS: DEXDEVIC is not summarized. DEXDEVIC is used to determine a unique key for records within a file based on the time-span, so other elements in the file can be summarized.

USAGE NOTES: DEXDEVIC is unique only within the context of the collection node (SYSID). Typical values are DUA0, DUA1, and so on.

D E X D U R Requested Recording Interval Time

The requested interval time in seconds specified to the VMS MONITOR.

TYPE: Common accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: DEXTS - Monitor Time Stamp
SPRCLSDR - CLS CLUSTER Record Duration (Type 19)
SPRDLKDR - DLK DLOCK Record Duration (Type 14)
SPRDNTDR - DNT DECNET Record Duration (Type 8)
SPRFCPDR - FCP Record Duration (Type 5)
SPRFSCDR - FSC CACHE Record Duration (Type 11)
SPRIOVDR - IOV IO Record Duration (Type 4)
SPRLOKDR - LOK LOCK Record Duration (Type 7)
SPRMSCDR - MSC MSCP SERVER Record Duration (Type21)
SPRPGEDR - PGE PAGE Record Duration (Type 3)
SPRPOLDR - POL POOL Record Duration (Type 6)
SPRSTADR - STA STATES Record Duration (Type 1)
SPRSYTDR - SYT SYSTEM Record Duration (Type 17)

ELEMENT DERIVATION: DEXDUR is obtained from the VMS MONITOR FILE HEADER record (type 128) from MNR_HDR$L_INTERVAL.

RANGE OF VALUE:

SUMMARIZATION PROCESS: DEXDUR contains the sum of all occurrences of DEXDUR that were included in the level of summarization.

USAGE NOTES: Note that DEXDUR is a constant value for a given monitor recording request. In contrast, the value of ENDTS minus STARTTS will vary depending on the actual timestamps of each record written by the monitor.

D E X G R P — User Identification Code Group

DEXGRP contains the high-order word of the user identification code for the owner of the process.

TYPE: Retained - Numeric, measurement unit is octal integer, 2 bytes long.

SEE ALSO: DEXMBR - User Identification Code Member
PROGRP - User Identification Code Group

ELEMENT DERIVATION: DEXGRP is set to the value of the ACR$W_GRP field in the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: DEXGRP takes an octal integer value from 1 to 77777.

SUMMARIZATION PROCESS: DEXGRP contains the last value of DEXGRP processed in the level of summarization.

USAGE NOTES: Group names are assigned by the VAX system manager. By itself, DEXGRP represents a class of VMS system users; when combined with DEXMBR, it is a unique identifier for an authorized account. If the value is between 40000 and 77777, DEXGRP indicates that a resource identifier, rather than a user account is the owner of the process. When the process owner is a resource identifier, DEXMBR will be 177777.

DEXGRP can be used to summarize detail records by group number. This summarization can show whether certain groups of users use resources at different levels from other groups.

D E X I N T N O — Reserved Field Incremented by Monitor

An incremental value observed in VMS MONITOR CLASS record headers to increase by one with each successive interval.

TYPE: Common retained data element - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXINTNO is obtained from each of the VMS MONITOR CLASS records, in the CLASS HEADER portion of each record. Data is from the MNR_CLS$W_RESERVED field.

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DEXINTNO contains the value of the last occurrence of DEXINTNO in the observations processed for the summarization level.

USAGE NOTES: Used in the DAILY input processing to match detail observations by recording interval.

D E X J O B — Jobname

DEXJOB contains the jobname assigned when the first process in the job group was initiated.

TYPE: Common, Retained - Character, length 39.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXJOB is set to the value of the ACR$W_JOBNAME field in the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: DEXJOB can contain any combination of 1 to 39 alphanumeric characters, including the symbols '_' and '$'.

SUMMARIZATION PROCESS: DEXJOB contains the last value of DEXJOB processed in the level of summarization.

USAGE NOTES: None

D E X J O B I D | Job ID

DEXJOBID contains the job ID of the job under which the process ran.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXJOBID is set to the value of the ACR$L_JOBID field in the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: DEXJOBID takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: DEXJOBID contains the last value of DEXJOBID processed in the level of summarization.

USAGE NOTES: The value stored in DEXJOBID is set by the first process started in the process tree at the interactive, batch or remote process level. Any subprocesses or detached processes started will receive a separate process ID but will keep the same job ID as the parent process. DEXJOBID is unique at a given time, but VMS will reuse old values when necessary.

D E X M B R | User Identification Code Member

DEXMBR contains the low-order word of the user identification code for the owner of the process.

TYPE: Retained - Numeric, measurement unit is octal integer, 2 bytes long.

SEE ALSO: DEXGRP - User Identification Code Group
PROMBR - User Identification Code Member

ELEMENT DERIVATION: DEXMBR is set to the value of the ACR$W_MEM field in the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: PROMBR takes an octal integer value from 1 to 177777.

SUMMARIZATION PROCESS: DEXMBR contains the last value of DEXMBR processed in the level of summarization.

USAGE NOTES: Member names are set up by the VAX system manager. When combined with the group name, the member makes up the user identification code. The user identification code is an identifier for authorized user accounts. If DEXMBR is 177777, then the owner is a resource identifier rather than a user account.

DEXMBR should be used in conjunction with DEXGRP to uniquely identify users because several unrelated user accounts may share the same member name. DEXMBR and DEXGRP combined may be used to search the DETAIL file for information relating to individual user accounts.

D E X M O N V R | VMS Monitor Version

The version number of the VMS Monitor software.

TYPE: Common data element - Character, length 4.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXMONVR is obtained from the VMS MONITOR FILE HEADER record (type 128) from MNR_HDR$T_IDENT.

RANGE OF VALUE: This element may have any of the following values:

o '4.7' when IDENT = 'MON20044'
o '5.1' when IDENT = 'MON25050'
o '5.3' when IDENT = 'MON27050'
o '5.4' when IDENT = 'MON29050'
o '6.0' when IDENT = 'MON30050'

SUMMARIZATION PROCESS: DEXMONVR contains the value of the last occurrence of DEXMONVR in the observations processed for the summarization level.

USAGE NOTES: None

D E X N A D D R | Node Address

DEXNADDR contains the node address for the remote node that caused this process to begin.

TYPE: Retained - Character, length 16.

SEE ALSO: DEXNODE - Nodename

ELEMENT DERIVATION: DEXNADDR is set to the value of the ACR$W_NODEADDR field in the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: DEXNADDR can contain any combination of 0 to 16 alphanumeric characters, including the symbols '_' and '$'.

SUMMARIZATION PROCESS: DEXNADDR contains the last value of DEXNADDR processed in the level of summarization.

USAGE NOTES: DEXNADDR can be used to report resource usage by specific remote addresses. If the process is run locally, the node address is empty. The node address can be derived from the formula:

node address = (area number * 1024) + node number

The area number and node number are stored in the network database and are available by using NCP.

D E X N O D E | Nodename

DEXNODE contains the name of the node of the remote process.

TYPE: Retained - Character, length 16.

SEE ALSO: DEXNADDR - Node Address

ELEMENT DERIVATION: DEXNODE is set to the value of the string pointed to by the ACR$W_NODENAME field of the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: DEXNODE can contain any combination of 1 to 16 alphanumeric characters, including the symbols '_' and '$'.

SUMMARIZATION PROCESS: DEXNODE contains the last value of DEXNODE processed in the level of summarization.

USAGE NOTES: DEXNODE has a one-to-one correspondence with DEXNADDR so one of these fields could be considered redundant.

D E X O W N I D | Owner ID

DEXOWNID contains the process ID number of the process that owns this subprocess.

TYPE: Retained - Numeric, measurement unit is hexadecimal integer, 4 bytes long.

SEE ALSO: DEXPID - Process ID

ELEMENT DERIVATION: DEXOWNID is set to the value of the ACR$L_OWNER field in the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: PROEID takes any hexadecimal numeric value in the range from 00000000 to FFFFFFFF.

SUMMARIZATION PROCESS: DEXOWNID contains the last value of DEXOWNID processed in the level of summarization.

USAGE NOTES: If this is a top-level process, DEXOWNID is zero. If this is a subprocess, DEXOWNID identifies the owning process. Process ID numbers are reused as needed by VMS. To find the owner or parent of a subprocess, match the DEXOWNID with the DEXPID.

```
+-----------------+
| D E X P I D     |  Process ID
+-----------------+
DEXPID contains the external ID (EPID) number of the process.

TYPE:  Retained - Numeric, measurement unit is hexadecimal
integer, 4 bytes long.

SEE ALSO:  PROPID - Process ID

ELEMENT DERIVATION:  DEXPID is set to the value of the
ACR$L_PID field in the ACR$K_ID packet of the VMS ACCOUNTING
records.

RANGE OF VALUE:  DEXPID takes any hexadecimal numeric value
in the range from 00000001 to FFFFFFFF.

SUMMARIZATION PROCESS:  DEXPID contains the last value of
DEXPID processed in the level of summarization.

USAGE NOTES:  Process ID is assigned to the process by the
system when:

o  Users LOGIN directly
o  Batch jobs start
o  Network services are invoked
o  Subprocesses are created
o  Detached processes are created with 'RUN/DETACHED'

There are never two processes on the system at the same time
that have the same DEXPID.  The system may reuse the number
after the process terminates.

During construction of the DESPRX file, the DEXPID value from
the DEAPRC file is matched with the PROEID from the DEMPRO
file.  The DEXPID element is kept in the DESPRX file to store
this common key value for external process ID (EPID).


+-----------------+
| D E X P R I V 1 |  Privilege 1 Mask
+-----------------+
DEXPRIV1 contains the mask of the first 32 VMS privileges.

TYPE:  Retained - Numeric, measurement unit is hexadecimal
integer, 4 bytes long.

SEE ALSO:  DEXPRIV2 - Privilege 2 Mask

ELEMENT DERIVATION:  DEXPRIV1 is set to the value of the
first 4 bytes of the ACR$Q_PRIV field in the ACR$K_ID packet
of the VMS ACCOUNTING records.

RANGE OF VALUE:  DEXPRIV1 takes any hexadecimal numeric value
in the range from 00000000 to FFFFFFFF.

SUMMARIZATION PROCESS:  DEXPRIV1 contains the last value of
DEXPRIV1 processed in the level of summarization.

USAGE NOTES:  Each bit represents one of the VMS privileges;
if the bit is set, the privilege is enabled, otherwise, it is
disabled.  The default VMS privileges NETMBX and TMPMBX have
a mask of 00108000 in DEXPRIV1.  Processes should be run with
as few privileges enabled as possible so there is less
opportunity to cause harm inadvertently.  If a process
terminates and DEXPRIV1 is FFFFFFFF, all privileges were
enabled.  When this occurs, the programmer or VAX system
manager should investigate the appropriate privilege usage.
Processes that run with all privileges enabled expose VMS to
security threat.


+-----------------+
| D E X P R I V 2 |  Privilege 2 Mask
+-----------------+
DEXPRIV2 contains the mask of the last 32 VMS privileges.

TYPE:  Retained - Numeric, measurement unit is hexadecimal
integer, 4 bytes long.

SEE ALSO:  DEXPRIV1 - Privilege 1 Mask

ELEMENT DERIVATION:  DEXPRIV2 is set to the value of the last
4 bytes of the ACR$Q_PRIV field in the ACR$K_ID packet of the
VMS ACCOUNTING records.

RANGE OF VALUE:  DEXPRIV2 takes any hexadecimal numeric value
in the range from 00000000 to FFFFFFFF.

SUMMARIZATION PROCESS:  DEXPRIV2 contains the last value of
DEXPRIV2 processed in the level of summarization.

USAGE NOTES:  Each bit represents one of the VMS privileges;
if the bit is set, the privilege is enabled, otherwise, it is
disabled.  The default VMS privileges NETMBX and TMPMBX have
a mask of 00000000 in DEXPRIV2.  Processes should be run with
as few privileges enabled as possible so there is less
opportunity to cause harm inadvertently.  If a process
terminates and DEXPRIV2 is FFFFFFFF, all privileges were
enabled.  When this occurs, the programmer or VAX system
manager should investigate the appropriate privilege usage.
Processes that run with all privileges enabled expose VMS to
```

severe security threats.

| D E X P R T Y | Priority

DEXPRTY contains the priority at which the process ran.

TYPE: Retained - Numeric, measurement unit is integer, 1 byte long.

SEE ALSO: PROPRTY - Priority

ELEMENT DERIVATION: DEXPRTY is set to the value of the ACR$B_PRI field in the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: DEXPRTY takes any integer value from 0 to 30.

SUMMARIZATION PROCESS: DEXPRTY contains the last value of DEXPRTY processed in the level of summarization.

USAGE NOTES: DEXPRTY is the actual priority level subtracted from 31. If DEXPRTY is 15 or less, the process ran in real-time mode and was not preempted by the scheduler. On most systems, SWAPPER is the only process that runs at a high priority; its DEXPRTY is 15.

| D E X Q U E U E | Queue Name

DEXQUEUE contains the name of the queue associated with a batch or print job.

TYPE: Common, Retained - Character, length 39.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXQUEUE is set to the value of the string pointed to by the ACR$W_QUEUE field in the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: DEXQUEUE can contain any combination of 0 to 39 alphanumeric characters, including the symbols '_' and '$'.

SUMMARIZATION PROCESS: DEXQUEUE contains the last value of DEXQUEUE processed in the level of summarization.

USAGE NOTES: If the process is not batch or print, DEXQUEUE is empty. If the process is for print, the queue name can be used to determine the destination of the print output.

| D E X R M T I D | Remote ID

DEXRMTID contains the ID of the process on the remote node that initiated the local process.

TYPE: Retained - Character, length 16.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXRMTID is set to the value of the string pointed to by the ACR$W_REMOTEID field in the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: DEXRMTID can contain any combination of 0 to 16 alphanumeric characters, including the symbols '_' and '$'.

SUMMARIZATION PROCESS: DEXRMTID contains the last value of DEXRMTID processed in the level of summarization.

USAGE NOTES: The string in this field varies widely, depending on the network protocol used. Under DECnet, DEXRMTID is always the VMS user name on the remote node.

DEXRMTID and either DEXNADDR or DEXNODE can be used to identify the individual responsible for starting the process on the system.

| D E X S T A T S | Status

DEXSTATS contains the final status of the last image run by the process.

TYPE: Retained - Character, 8 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXSTATS is set from the value of the ACR$L_STATUS field in the ACR$K_RESOURCE packet of the VMS ACCOUNTING records. This input field is a hexadecimal integer, 4 bytes long, with a value in the range from

00000001 to FFFFFFFF. DEXSTATS is assigned as follows:

```
SELECT (MOD(ACR$L_STATUS,8));
 WHEN (0) DEXSTATS='WARNING ';
 WHEN (1) DEXSTATS='SUCCESS ';
 WHEN (2) DEXSTATS='ERROR   ';
 WHEN (3) DEXSTATS='INFO    ';
 WHEN (4) DEXSTATS='FATAL   ';
 OTHERWISE DEXSTATS='UNKNOWN ';
END;
```

RANGE OF VALUE: DEXSTATS takes one of the following values: "WARNING ","SUCCESS ","ERROR ","INFO ","FATAL ", or "UNKNOWN ".

SUMMARIZATION PROCESS: DEXSTATS contains the last value of DEXSTATS processed in the level of summarization.

USAGE NOTES: If the final status in the original input field is an odd number, it signals successful completion; even numbers indicate a degree of unsuccessful completion. If the final status is a 2 or A, it indicates a warning. If the final status is a 4 or C, a severe error occurred and the problem should be investigated. The most common successful status is 00010001.

D E X S U S T B — DEXSUS System Boot Time Stamp

DEXSUSTB contains the date and time of the last boot of the system.

TYPE: Common, maximum - Numeric, measurement unit is SAS datetime value.

SEE ALSO: DEXBOOTS - Monitor System Boot Time Stamp
SYUUPTIM - System Uptime

ELEMENT DERIVATION: DEXSUSTB is set to the value of the BOOTTIME field from the DEXSUS SYI3 separator record.

RANGE OF VALUE: Any valid SAS time-stamp value. The value of this data element is not affected by your use of the 13MONTHYEAR option in prefix.MICS.PARMS(SITE).

SUMMARIZATION PROCESS: DEXSUSTB contains the maximum value of DEXSUSTB processed for the level of summarization.

USAGE NOTES: None

D E X S U S T S — DEXSUS Collect Time Stamp

DEXSUSTS contains the time stamp from the DEXSUS collect routine for the specific source record associated with each unique file.

TYPE: Common, maximum - Numeric, measurement unit is SAS datetime value.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXSUSTS is usually set to the value of the time stamp reported on DEXSUS System Information record from the SHOW SYSTEM command output. One exception is the case of DEXSUS DISK USAGE records from Analyze/Disk, which provide their own collect time stamp. DEXSUSTS is set to this value for the DKD and DKU files.

RANGE OF VALUE: Any valid SAS time-stamp value. The value of this data element is not affected by your use of the 13MONTHYEAR option in prefix.MICS.PARMS(SITE).

SUMMARIZATION PROCESS: DEXSUSTS contains the maximum value of DEXSUSTS processed for the level of summarization.

USAGE NOTES: The DEXSUSTS is normally equal to ENDTS for most record types in the System Usage information area. Where a specific source record provides an additional time stamp DEXSUSTS carries this value.

D E X T E R M — Terminal Name

DEXTERM contains the identification of the terminal associated with the process.

TYPE: Retained - Character, length 16.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXTERM is set to the value of the ACR$W_TERMINAL field in the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: DEXTERM can contain any combination of 0 to 16 alphanumeric characters, including the symbols '_' and '$'.

SUMMARIZATION PROCESS: DEXTERM contains the last value of

DEXTERM processed in the level of summarization.

USAGE NOTES: If the process is a batch job or a detached process, this element is empty. The terminal identification is established by VMS at process creation and does not change throughout the life of the process.

| D E X T S | Monitor Time Stamp

DEXTS is the system time stamp at which each VMS MONITOR CLASS record is recorded.

TYPE: Common retained data element - Numeric, measurement unit is SAS time value.

SEE ALSO:
- STARTTS - Start Time Stamp
- ENDTS - End Time Stamp
- DURATION - Length of Recording Interval
- DEXDUR - Requested Recording Interval Time
- INTERVLS - Number of Recording Intervals

ELEMENT DERIVATION: DEXTS is obtained from each of the VMS MONITOR CLASS records in the CLASS HEADER portion of each record. Data is from the MNR_CLS$Q_STAMP field.

RANGE OF VALUE: DEXTS contains any valid SAS time stamp value. The value of this data element is not affected by use of the 13MONTHYEAR option in prefix.MICS.PARMS(SITE).

SUMMARIZATION PROCESS: DEXTS contains the value of the last occurrence of DEXTS in the observations processed for the summarization level.

USAGE NOTES: Note that while DEXDUR is a constant value for a given VMS MONITOR recording request, DURATION (ENDTS-STARTTS) will vary depending on the actual time stamps (DEXTS) of each record written by VMS MONITOR.

For all VMS Monitor information area files, ENDTS is obtained from the earliest value of DEXTS found in each interval of detail input data.

| D E X U S E R | Username

DEXUSER contains the username of the process creator.

TYPE: Retained - Character, length 16.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXUSER is set to the value of ACR$W_USERNAME field in the ACR$K_ID packet of the VMS ACCOUNTING records.

RANGE OF VALUE: DEXUSER can contain any combination of 1 to 16 alphanumeric characters, including the symbols '_' and '$'.

SUMMARIZATION PROCESS: DEXUSER contains the last value of DEXUSER processed in the level of summarization.

USAGE NOTES: Username matches the name maintained in the user authorization file. When a process is started from a remote node, DEXUSER is the username by which the remote node connects and gains resources.

| D E X V O L N M | Volume Name

DEXVOLNM contains the volume name of relative volume 1.

TYPE: Common, retained - Character, length 12.

SEE ALSO: DKDVSNAM - Volume Set Name

ELEMENT DERIVATION: DEXVOLNM is set to the value of the USG$T_VOLNAME field of the Analyze/Disk Identification Record (type 1).

RANGE OF VALUE: DEXVOLNM can contain any combination of 1 to 12 alphanumeric characters.

SUMMARIZATION PROCESS: DEXVOLNM is not summarized. DEXVOLNM is used to determine a unique key for records within a file based on the time-span, so other elements in the file can be summarized.

USAGE NOTES: None

```
+-----------------+
| D E X V O L S R |  Volume Serial Number
+-----------------+
DEXVOLSR contains the serial number of the disk volume.

TYPE:  Common, retained - Numeric, measurement unit is an
octal longword (4 byte) value.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  DEXVOLSR is set to the value of the
USG$L_SERIALNUM field of the Analyze/Disk Identification
Record (type 1).

RANGE OF VALUE:  DEXVOLSR takes any octal numeric value that
ranges from a minimum of 0 to a maximum that is limited only
by practical bounds.

SUMMARIZATION PROCESS:  DEXVOLSR contains the last value of
DEXVOLSR processed in the level of summarization.

USAGE NOTES:  None
```

B.2 - VAX/VMS Accounting Information (DEA) Area

The following sections describe the VAX/VMS Accounting information area (DEA). The statistics for these files are obtained from the VMS ACCOUNTING data.

1 - File Relationships and Common Data Elements
2 - The VMS Accounting Process Termination (DEAPRC) File
3 - The VMS Accounting Image Termination (DEAIMG) File
4 - The VMS Accounting LOGIN Failed (DEA_LF) File
5 - The VMS Accounting Print Queued (DEAPRQ) File
6 - The VMS Accounting Initialization (DEAINT) File

B.2.1 - File Relationships and Common Data Elements

Not available at this time.

B.2.2 - The VMS Accounting Process Termination (DEAPRC) File

FILE ABSTRACT: VMS Accounting Process Termination File	
CONTENT	Quantifies the resource usage of each process on the VMS system during the monitoring interval.
USE	To report resource usage of each process.
SOURCE	VMS ACCOUNTING record type 01 as documented in the VMS Accounting Reference Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	DAY	HOUR	ENDTS	
DAYS	N/A				
WEEKS	N/A				
MONTHS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	ZONE			
YEARS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	ZONE				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
X.....E	DAY	- Day of Month
X..MY.E	DEXACT1	- DIVISION
X..MY.E	DEXACT2	- UIC GROUP
X..MY.E	DEXACT3	- USER NAME
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..MY.E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year
X..MY.E	YEAR	- Year of Century
X..MY.E	ZONE	- Time Zone
Common Data Elements		
X.....E	DAYNAME	- Name of Day of Week
X.....E	DEXACCNT	- Account Name

```
 X.....E  DEXGRP   - User Identification Code Group
 X.....E  DEXJOB   - Jobname
 X.....E  DEXJOBID - Job ID
 X.....E  DEXMBR   - User Identification Code Member
 X.....E  DEXNADDR - Node Address
 X.....E  DEXNODE  - Nodename
 X.....E  DEXOWNID - Owner ID
 X.....E  DEXPID   - Process ID
 X.....E  DEXPRIV1 - Privilege 1 Mask
 X.....E  DEXPRIV2 - Privilege 2 Mask
 X.....E  DEXPRTY  - Priority
 X.....E  DEXQUEUE - Queue Name
 X.....E  DEXRMTID - Remote ID
 X.....E  DEXSTATS - Status
 X.....E  DEXTERM  - Terminal Name
 X.....E  DEXUSER  - Username
 X..MY.E  DURATION - Recording Interval Time
 X..MY.E  ENDTS    - End Time Stamp
 X.....E  ORGSYSID - Originating System Identification
 X..MY.E  STARTTS  - Start Time Stamp

Retained Data Elements

 X.....E  PRCTYPE  - Process Type

Accumulated Data Elements

 X..MY.E  PRCBUFIO - Buffered I/Os
 X..MY.E  PRCCOST  - Processing Charges
 X..MY.E  PRCCPUNI - Number of Instructions Executed
 X..MY.E  PRCCPUTM - Total Process CPU Time
 X..MY.E  PRCDIRIO - Direct I/Os
 X..MY.   PRCFLTIO - Page Fault I/Os
 X..MY.E  PRCIMAGE - Image Count
 X..MY.E  PRCPAGFL - Page File Peak Usage
 X..MY.   PRCPAGFT - Page Faults
 X..MY.E  PRCPRCNT - Process Count
 X..MY.E  PRCVOLS  - Volumes Mounted

Maximum Data Elements

 X..MY.   PRCWKSPK - Working Set Peak
              File

+-----------------+
| P R C B U F I O |  Buffered I/Os
+-----------------+
PRCBUFIO contains the number of buffered I/Os requested by
the process.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  IMGBUFIO - Buffered I/Os

ELEMENT DERIVATION:  PRCBUFIO is set to the value of the
ACR$L_BIOCNT field of the ACR$K_RESOURCES packet of the VMS
ACCOUNTING PROCESS record.

RANGE OF VALUE:  PRCBUFIO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  PRCBUFIO contains the sum of all
values of PRCBUFIO processed in the level of summarization.

USAGE NOTES:  PRCBUFIO reflects the number of buffered I/O
operations, not the size of those operations.  Buffered I/O
operations usually do not affect system performance much.
Systems with all terminals, printers and other buffered
devices under direct control of the CPU can be affected by
large numbers of buffered I/Os.

+-----------------+
| P R C C O S T   |  Processing Charges
+-----------------+
PRCCOST contains the total processing costs for a process.
The value of PRCCOST is zero unless a user computation is
performed or the file containing this element is selected for
billing processing by the CA MICS Accounting and Chargeback
component.

TYPE:  Computed data element - Numeric, measurement unit is
real number.

SEE ALSO:  IMGCOST - Processing Charges
           PRQCOST - Processing Charges

ELEMENT DERIVATION:  In the DETAIL timespan, PRCCOST is set
by the CA MICS Accounting and Chargeback component interface.

RANGE OF VALUE:  The value of this element is determined by
the charging algorithms in effect at your installation.

SUMMARIZATION PROCESS:  PRCCOST contains the sum of all
occurrences of PRCCOST that were included in the level of
summarization.

USAGE NOTES:  None
```

```
+-----------------+
| P R C C P U N I |  Number of Instructions Executed
+-----------------+
PRCCPUNI contains an estimate of the total number of CPU
instructions executed for a process, calculated by
multiplying the measured CPU time by a user-supplied VUPS
(where VAX Units of Power for VAX 11/780=1) rate.

TYPE:  Accumulated data element - Numeric, measurement unit
is integer.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  CPUTM*MIPS

RANGE OF VALUE:  The value of this element ranges from a
minimum of 0 to a maximum that is limited only by practical
bounds.

SUMMARIZATION PROCESS:  PRCCPUNI contains the sum of all
occurrences of PRCCPUNI that were included in the level of
summarization.

USAGE NOTES:  PRCCPUNI is an approximation based on system
engineering concepts that were valid for second-generation
equipment with hard-wired instructions and single-threaded
execution.  The accuracy of this element is a function of the
instruction mix used in the program, the contending load (if
any) using the system concurrently with the step, the
processor architecture, the system microcode level, and the
MIPS rate supplied by the user during the CA MICS generation
process.  Altering any of these factors will change the
reported value of PRCCPUNI for replicate runs, even though
the exact code executes each time a step is initiated.

When used with caution, PRCCPUNI can be helpful in comparing
the relative efficiency of different software realizations of
the same function (e.g., the relative path lengths of two
data base management systems for a backup/reload cycle
performed on equivalent collections of data).


+-----------------+
| P R C C P U T M |  Total Process CPU Time
+-----------------+
PRCCPUTM contains the number of seconds during which the
process had control of the CPU.

TYPE:  Accumulated - Numeric, measurement unit is SAS time
value, 4 bytes long.

SEE ALSO:  SPRCPUTM - System CPU Busy Time
           IMGCPUTM - Total Image CPU Time
           PROCPUTM - CPU Time

ELEMENT DERIVATION:  PRCCPUTM is set to the value of the
ACR$L_CPUTM field from the ACR$K_RESOURCE packet of the VMS
ACCOUNTING PROCESS record.

RANGE OF VALUE:  PRCCPUTM is a SAS time value ranging from
00.00.00 to the size of the DURATION (measurement interval)
value.

SUMMARIZATION PROCESS:  PRCCPUTM contains the sum of all
values of PRCCPUTM processed in the level of summarization.

USAGE NOTES:  For most processes, PRCCPUTM includes time
spent in user and supervisor modes only.  If the system
spends CPU time in kernel or executive modes on behalf of a
process, but not under the process's direct control, the time
is not recorded in PRCCPUTM field.  For example, if the
process requests a system service that the system must change
to kernel mode to provide, the CPU time used to honor the
request will not be recorded in PRCCPUTM.


+-----------------+
| P R C D I R I O |  Direct I/Os
+-----------------+
PRCDIRIO contains the number of direct I/Os requested by the
process.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  IMGDIRIO - Direct I/Os

ELEMENT DERIVATION:  PRCDIRIO is set to the value of the
ACR$L_DIOCNT field in the ACR$K_RESOURCES packet of the VMS
ACCOUNTING PROCESS record.

RANGE OF VALUE:  PRCDIRIO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  PRCDIRIO contains the sum of all
values of PRCDIRIO processed in the level of summarization.

USAGE NOTES:  PRCDIRIO reflects the number of direct I/O
operations, not the size of those operations.  Direct I/Os
can have a great effect on system performance, but the effect
varies widely depending on the average transaction size.  If
the number of direct I/Os is constant, but average
```

transaction size is larger, the negative effect will be greater.

P R C F L T I O Page Fault I/Os

PRCFLTIO contains the number of fault I/Os generated by the process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRCPAGFT - Page Faults
IMGFLTIO - Page Fault I/Os

ELEMENT DERIVATION: PRCFLTIO is set to the value of the ACR$L_FAULTIO field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING PROCESS record.

RANGE OF VALUE: PRCFLTIO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: PRCFLTIO contains the sum of all values of PRCFLTIO processed in the level of summarization.

USAGE NOTES: PRCFLTIO is the sum of the page fault read I/Os and the page fault write I/Os that occurred on behalf of the process. Because the memory manager tries to cluster several page faults into a single I/O operation, PRCFLTIO is usually smaller than PRCPAGFL.

P R C I M A G E Image Count

PRCIMAGE contains the number of images activated by the process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: IMGIMAGE - Image Sequence Number

ELEMENT DERIVATION: PRCIMAGE is set to the value of the ACR$L_IMGCNT field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING PROCESS record.

RANGE OF VALUE: PRCIMAGE takes any integer value from 1 to 2,147,483,647.

SUMMARIZATION PROCESS: PRCIMAGE contains the sum of all values of PRCIMAGE processed in the level of summarization.

USAGE NOTES: Processes are a set of images run in a time-span started with a LOGIN and terminated with a LOGOUT. PRCIMAGE is the number of images run by the process before it terminated.

P R C P A G F L Page File Peak Usage

PRCPAGFL contains the maximum number of pages used by the process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: IMGPAGFL - Page File Peak Usage

ELEMENT DERIVATION: PRCPAGFL is set to the value of the ACR$L_PAGEFL field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING PROCESS record.

RANGE OF VALUE: PRCPAGFL takes any integer value from 0 to the largest value of PGFLQUO for any user on the system.

SUMMARIZATION PROCESS: PRCPAGFL contains the sum of all values of PRCPAGFL processed in the level of summarization.

USAGE NOTES: The number of pages used by a process can be an indicator of the total effect on the system. Large processes tend to increase page faulting and swapping, which bottlenecks the I/O subsystem and steals cycles from the CPU. Large processes usually do more processing, although the effect on the CPU is usually mitigated by the large number of I/Os that they usually do. When looking for ways to structure applications differently, look to those that have a larger than average PRCPAGFL. Trimming images to be smaller should be done with care, however. The overhead of image activation could easily overcome any gains made from smaller page file usage.

```
+-----------------+
| P R C P A G F T |  Page Faults
+-----------------+
PRCPAGFT contains the number of page faults generated by the
process.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  IMGPAGFT - Page Faults
           PRCFLTIO - Page Fault I/Os

ELEMENT DERIVATION:  PRCPAGFT is set to the value of the
ACR$L_FAULTS field in the ACR$K_RESOURCES packet of the VMS
PROCESS ACCOUNTING PROCESS record.

RANGE OF VALUE:  PRCPAGFT takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  PRCPAGFT contains the sum of all
values of PRCPAGFT processed in the level of summarization.

USAGE NOTES:  A high value in PRCPAGFT means that the system
did not find the pages for the process in the processor's
physical memory when it needed them.  Some page faulting is
necessary for the images to be activated and initialize data
space.  For large images, however, much of the page faulting
may be avoided by adjusting the working set size.

Large amounts of page faulting affect the performance of the
process and overall system, in particular, hard faults in
PRCFLTIO.  Because the memory manager tries to cluster
several page faults into a single I/O operation, PRCFLTIO is
usually smaller than PRCPAGFT.


+-----------------+
| P R C P R C N T |  Process Count
+-----------------+
PRCPRCNT contains a count of the number of VMS Accounting
Process Termination records represented by the observation.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS Accounting Process Termination
is processed in DETAIL data, PRCPRCNT is set to a value of
1.

RANGE OF VALUE: PRCPRCNT takes an integer value from 1 to
a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: PRCPRCNT contains the sum of all
values of PRCPRCNT processed in the level of summarization.

USAGE NOTES: PRCPRCNT can be used as the divisor for
calculating per-process averages and ratios at any given
summarization level.


+-----------------+
| P R C T Y P E   |  Process Type
+-----------------+

The process type for remote process.

TYPE:  Common data element - Character, length 8.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  PRCTYPE is obtained from the VMS
ACCOUNTING packet header from ACR$V_SUBTYPE.

RANGE OF VALUE:  This element may contain any of the
following values:

    PRINTJOB - Print Job
    INTERACT - Interactive Process
    SUBPROC  - SUB Process
    DETACHED - Detached Process
    BATCH    - Batch Process
    NETWORK  - Network Process
    UNKNOWN  - Unknown

SUMMARIZATION PROCESS:  PRCTYPE contains the value of the
last occurrence of PRCTYPE in the observations processed for
the summarization level.

USAGE NOTES:  None


+-----------------+
| P R C V O L S   |  Volumes Mounted
+-----------------+
PRCVOLS contains the number of volume mounts requested by the
process.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.
```

SEE ALSO: IMGVOLS - Volumes Mounted

ELEMENT DERIVATION: PRCVOLS is set to the value of the ACR$L_VOLUMES field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING PROCESS record.

RANGE OF VALUE: PRCVOLS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: PRCVOLS contains the sum of all values of PRCVOLS processed in the level of summarization.

USAGE NOTES: When an process needs to use tape drives, removable media disks, or allocated space on fixed disks, it gains exclusive use of that device by executing a mount request. PRCVOLS is the count of volume mount requests made by the process.

| P R C W K S P K | Working Set Peak

PRCWKSPK contains the maximum working set size of the process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: IMGWKSPK - Working Set Peak

ELEMENT DERIVATION: PRCWKSPK is set to the value of the ACR$L_WSPEAK field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING PROCESS record.

RANGE OF VALUE: PRCWKSPK takes any integer value from 1 to the smallest value specified by the SYSGEN parameter WSMAX or the user authorization file parameter WSEXTENT.

SUMMARIZATION PROCESS: PRCWKSPK contains the sum of all values of PRCWKSPK processed in the level of summarization.

USAGE NOTES: If PRCWKSPK is smaller than WSQUOTA for the user, all page faulting was a result of image activation. If PRCWKSPK is nearly as large as WSEXTENT, the process was borrowing pages from free memory. When PRCWKSPK is growing to nearly the size of WSEXTENT, it indicates that there is sufficient memory free and the system is not swapping. If paging is high and this condition occurs, adjusting WSQUOTA and WSEXTENT higher may improve overall performance.

FILE ABSTRACT: VMS Accounting Image Termination File	
CONTENT	Quantifies the resource usage of each image on the VMS system during the monitoring interval.
USE	To report resource usage of each image.
SOURCE	VMS ACCOUNTING record type 03 as documented in the VMS Accounting Reference Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	DAY	HOUR	ENDTS	
DAYS	N/A				
WEEKS	N/A				
MONTHS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	ZONE			
YEARS	N/A				
TABLES	N/A				

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
	Sequence/Summary Data Elements	
X.....E	DAY	- Day of Month
X..M..E	DEXACT1	- DIVISION
X..M..E	DEXACT2	- UIC GROUP
X..M..E	DEXACT3	- USER NAME
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year
X..M..E	YEAR	- Year of Century
X..M..E	ZONE	- Time Zone
	Common Data Elements	
X.....E	DAYNAME	- Name of Day of Week
X.....E	DEXACCNT	- Account Name
X.....E	DEXGRP	- User Identification Code Group

```
 X.....E  DEXJOB   - Jobname
 X.....E  DEXJOBID - Job ID
 X.....E  DEXMBR   - User Identification Code Member
 X.....E  DEXNADDR - Node Address
 X.....E  DEXNODE  - Nodename
 X.....E  DEXOWNID - Owner ID
 X.....E  DEXPID   - Process ID
 X.....E  DEXPRIV1 - Privilege 1 Mask
 X.....E  DEXPRIV2 - Privilege 2 Mask
 X.....E  DEXPRTY  - Priority
 X.....E  DEXQUEUE - Queue Name
 X.....E  DEXRMTID - Remote ID
 X.....E  DEXSTATS - Status
 X.....E  DEXTERM  - Terminal Name
 X.....E  DEXUSER  - Username
 X..M..E  DURATION - Recording Interval Time
 X..M..E  ENDTS    - End Time Stamp
 X.....E  ORGSYSID - Originating System Identification
 X..M..E  STARTTS  - Start Time Stamp

Retained Data Elements

 X.....   IMGIMAGE - Image Sequence Number
 X.....E  IMGNAME  - Image Name
 X.....E  IMGTYPE  - Process Type

Accumulated Data Elements

 X..M..E  IMGBUFIO - Buffered I/Os
 X..M..E  IMGCOST  - Processing Charges
 X..M..E  IMGCPUNI - Number of Instructions Executed
 X..M..E  IMGCPUTM - Total Image CPU Time
 X..M..E  IMGDIRIO - Direct I/Os
 X..M..   IMGFLTIO - Page Fault I/Os
 X..M..E  IMGIMCNT - Image Count
 X..M..   IMGPAGFL - Page File Peak Usage
 X..M..   IMGPAGFT - Page Faults
 X..M..E  IMGVOLS  - Volumes Mounted

Maximum Data Elements

 X..M..   IMGWKSPK - Working Set Peak
              File


+-----------------+
| I M G B U F I O |  Buffered I/Os
+-----------------+
IMGBUFIO contains the number of buffered I/Os requested by
the image.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  PRCBUFIO - Buffered I/Os

ELEMENT DERIVATION:  IMGBUFIO is set to the ACR$L_BIOCNT
field of the ACR$K_RESOURCES packet of the VMS ACCOUNTING
IMAGE record.

RANGE OF VALUE:  IMGBUFIO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  IMGBUFIO contains the sum of all
values of IMGBUFIO processed in the level of summarization.

USAGE NOTES:  IMGBUFIO reflects the number of buffered I/O
operations, not the size of those operations.  Buffered I/O
operations usually do not affect system performance much.
Systems with all terminals, printers, and other buffered
devices under direct control of the CPU can be affected by
large numbers of buffered I/Os.


+-----------------+
| I M G C O S T   |  Processing Charges
+-----------------+

IMGCOST contains the total processing costs for an image.
The value of IMGCOST is zero unless a user computation is
performed or the file containing this element is selected for
billing processing by the CA MICS Accounting and Chargeback
Component.

TYPE:  Computed data element - Numeric, measurement unit is
real number.

SEE ALSO:  PRCCOST - Processing Charges
           PRQCOST - Processing Charges

ELEMENT DERIVATION:  In the DETAIL timespan, IMGCOST is set
by the Accounting and Chargeback Component.

RANGE OF VALUE:  The value of this element is determined by
the charging algorithms in effect at your installation.

SUMMARIZATION PROCESS:  IMGCOST contains the sum of all
occurrences of IMGCOST that were included in the level of
summarization.

USAGE NOTES:  None
```

| I M G C P U N I | Number of Instructions Executed

IMGCPUNI contains an estimate of the total number of CPU instructions executed for an image, calculated by multiplying the measured CPU time by a user-supplied VUPS (where VAX Units of Power for VAX 11/780=1) rate.

TYPE: Accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: CPUTM*MIPS

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: IMGCPUNI contains the sum of all occurrences of IMGCPUNI that were included in the level of summarization.

USAGE NOTES: IMGCPUNI is an approximation based on system engineering concepts that were valid for second-generation equipment with hard-wired instructions and single-threaded execution. The accuracy of this element is a function of the instruction mix used in the program, the contending load (if any) using the system concurrently with the step, the processor architecture, the system microcode level, and the MIPS rate supplied by the user during the CA MICS generation process. Altering any of these factors will change the reported value of IMGCPUNI for replicate runs, even though the exact code executes each time a step is initiated.

When used with caution, IMGCPUNI can be helpful in comparing the relative efficiency of different software realizations of the same function (e.g., the relative path lengths of two database management systems for a backup/reload cycle performed on equivalent collections of data).

| I M G C P U T M | Total Image CPU Time

IMGCPUTM contains the amount of the CPU time used by the image during the DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is SAS time value, 4 bytes long.

SEE ALSO: SPRCPUTM - System CPU Busy Time
PRCCPUTM - Total Process CPU Time
PROCPUTM - CPU Time

ELEMENT DERIVATION: IMGCPUTM is set to the value of the ACR$L_CPUTM field in the ACR$K_RESOURCE packet of the VMS ACCOUNTING IMAGE record.

RANGE OF VALUE: IMGCPUTM is a SAS time value ranging from 00:00.00 to the size of the DURATION (measurement interval).

SUMMARIZATION PROCESS: IMGCPUTM contains the sum of all values of IMGCPUTM processed in the level of summarization.

USAGE NOTES: For most processes, IMGCPUTM includes time spent in user and supervisor modes only. If CPU time is spent in kernel or executive mode on behalf of a process, but not under the process' direct control, the time is not recorded in this field for the process. For example, if the process requests a system service for which the system must change to kernel mode to provide, the time that the CPU used honoring the request will not be recorded in IMGCPUTM for the process.

| I M G D I R I O | Direct I/Os

IMGDIRIO contains the number of direct I/Os requested by the image.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRCDIRIO - Direct I/Os

ELEMENT DERIVATION: IMGDIRIO is set to the value of the ACR$L_DIOCNT field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING IMAGE record.

RANGE OF VALUE: IMGDIRIO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: IMGDIRIO contains the sum of all values of IMGDIRIO processed in the level of summarization.

USAGE NOTES: IMGDIRIO reflects the number of direct I/O operations, not the size of those operations. Direct I/Os can have great effect on system performance, but the effect varies widely with average transaction size. If the number of direct I/Os is constant, but average transaction size is

larger, the negative effect will be greater.

I M G F L T I O Page Fault I/Os

IMGFLTIO contains the number of fault I/Os generated by the image.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: IMGPAGFT - Page Faults
PRCFLTIO - Page Fault I/Os

ELEMENT DERIVATION: IMGFLTIO is set to the value of the ACR$L_FAULTIO field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING IMAGE record.

RANGE OF VALUE: IMGFLTIO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: IMGFLTIO contains the sum of all values of IMGFLTIO processed in the level of summarization.

USAGE NOTES: IMGFLTIO is the sum of the page fault read I/Os and the page fault write I/Os that the image caused to occur. Since the memory manager tries to cluster several page faults into a single I/O operation, IMGFLTIO is usually smaller than IMGPAGFT.

I M G I M A G E Image Sequence Number

IMGIMAGE contains the sequence number in the process of this image.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRCIMAGE - Image Count

ELEMENT DERIVATION: IMGIMAGE is set to the value of ACR$L_IMGCNT field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING IMAGE record.

RANGE OF VALUE: IMGIMAGE takes any integer value from 1 to 2,147,483,647.

SUMMARIZATION PROCESS: IMGIMAGE contains the last value of IMGIMAGE processed in the level of summarization.

USAGE NOTES: Processes in VMS are a set of images run in a time-span started with a LOGIN and terminated with a LOGOUT. IMGIMAGE is the cardinal number of the image within its process. For example, if the image is the 15th image activated by this process, IMGIMAGE will be 15. The next image will be 16 and so forth.

The number in IMGIMAGE is determined at image activation. If a process has more than one thread running, the records in the VMS ACCOUNTING file need not be in numerical order by IMGIMAGE, because those records are entered according to image termination time.

I M G I M C N T Image Count

IMGIMCNT contains a count of the number of VMS Accounting Image Termination records represented by the observation.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS Accounting Image Termination is processed in DETAIL data, IMGIMCNT is set to a value of 1.

RANGE OF VALUE: IMGIMCNT takes an integer value from 1 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: IMGIMCNT contains the sum of all values of IMGIMCNT processed in the level of summarization.

USAGE NOTES: IMGIMCNT can be used as the divisor for calculating per-image averages and ratios at any given summarization level.

I M G N A M E Image Name

IMGNAME contains the name of the image for which the record is recorded.

TYPE: Retained - Character, length 200.

SEE ALSO: Not applicable

ELEMENT DERIVATION: IMGNAME is set to the value of the string pointed to by the ACR$T_IMAGENAME field in the ACR$K_IMAGENAME packet of the VMS ACCOUNTING IMAGE record.

RANGE OF VALUE: IMGNAME can contain any combination of 1 to 200 alphanumeric characters, including the symbols '_', '$', ';', ':', '.', and left and right square brackets.

SUMMARIZATION PROCESS: IMGNAME contains the last value of IMGNAME processed in the level of summarization.

USAGE NOTES: Since IMGNAME is the path name of the executable file that contains the image, there is no guarantee that two different strings are not actually the same image. VMS allows for a single file to appear on more than one directory; therefore, depending on the string used to access an image file that appears in two directories, the same physical file can show different values in IMGNAME. Since logical names are fully translated before the image name is written, if two records have the same IMGNAME, they are indeed the same image file.

I M G P A G F L — Page File Peak Usage

IMGPAGFL contains the maximum number of pages of the PAGE file used by the image.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRCPAGFL - Page File Peak Usage

ELEMENT DERIVATION: IMGPAGFL is set to the value of the ACR$L_PAGEFL field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING IMAGE record.

RANGE OF VALUE: IMGPAGFL takes any integer value from 0 to the largest value of PGFLQUO for any user on the system.

SUMMARIZATION PROCESS: IMGPAGFL contains the sum of all values of IMGPAGFL processed in the level of summarization.

USAGE NOTES: The number of pages used by an image can be a good indicator of the total effect of the image. Larger images cause more page faulting and swapping, which has a negative effect on the I/O subsystem and potentially on the CPU. Larger images usually do more processing, although the effect on the CPU is usually mitigated by the large number of I/Os that large images usually do. When looking for ways to structure applications differently, look for those that have a larger than average IMGPAGFL. Trimming images to be smaller should be done with care, however. The overhead of image activation could easily overcome any gains made from smaller PAGE file usage.

I M G P A G F T — Page Faults

IMGPAGFT contains the number of page faults generated by the image.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRCPAGFT - Page Faults
IMGFLTIO - Page Fault I/Os

ELEMENT DERIVATION: IMGPAGFT is set to the value of the ACR$L_FAULTS field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING IMAGE record.

RANGE OF VALUE: IMGPAGFT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: IMGPAGFT contains the sum of all values of IMGPAGFT processed in the level of summarization.

USAGE NOTES: A high value for IMGPAGFT means that the image did not find the pages of data or program memory resident in the physical memory of the machine when it needed them. Some page faulting is necessary for the image to be activated and initialize its data space. For large images, however, much of the page faulting may be avoidable by adjusting the working set size.

Large amounts of page faulting will negatively affect image and system performance, particularly the hard faults in IMGFLTIO. Since the memory manager tries to cluster several page faults into a single I/O operation, IMGFLTIO is usually smaller than IMGPAGFT.

```
+-----------------+
| I M G T Y P E   |  Process Type
+-----------------+

The Process type for remote process.

TYPE:  Common data element - Character, length 8.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  IMGTYPE is obtained from the VMS
ACCOUNTING packet header from ACR$V_SUBTYPE.

RANGE OF VALUE:  This element may contain any of the
following values:

 o 'PRINTJOB'    /* 'PRINT JOB'             */
 o 'INTERACT'    /* 'INTERACTIVE PROCESS'   */
 o 'SUBPROC '    /* 'SUB PROCESS'           */
 o 'DETACHED'    /* 'DETACHED PROCESS'      */
 o 'BATCH   '    /* 'BATCH PROCESS'         */
 o 'NETWORK '    /* 'NETWORK PROCESS'       */
 o 'UNKNOWN '

SUMMARIZATION PROCESS:  IMGTYPE contains the value of the
last occurrence of IMGTYPE in the observations processed for
the summarization level.

USAGE NOTES:  None


+-----------------+
| I M G V O L S   |  Volumes Mounted
+-----------------+
IMGVOLS contains the number of volume mounts requested by the
image.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  PRCVOLS - Volumes Mounted

ELEMENT DERIVATION:  IMGVOLS is set to the value of the
ACR$L_VOLUMES field in the ACR$K_RESOURCES packet of the VMS
ACCOUNTING IMAGE record.

RANGE OF VALUE:  IMGVOLS takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  IMGVOLS contains the sum of all
values of IMGVOLS processed in the level of summarization.

USAGE NOTES:  When an image needs to use tape drives,
removable media disks, or allocated fixed disks, it gains
exclusive use of that device by executing a mount request.
IMGVOLS is the count of those requests by the image.


+-----------------+
| I M G W K S P K |  Working Set Peak
+-----------------+
IMGWKSPK contains the maximum working set size of the image.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  PRCWKSPK - Working Set Peak

ELEMENT DERIVATION:  IMGWKSPK is set to the value of the
ACR$L_WSPEAK field in the ACR$K_RESOURCES packet of the VMS
ACCOUNTING IMAGE record.

RANGE OF VALUE:  IMGWKSPK takes any integer value from 1 to
the smaller of:
  o The SYSGEN parameter WSMAX
  o The user authorization file parameter WSEXTENT

SUMMARIZATION PROCESS:  IMGWKSPK contains the sum of all
values of IMGWKSPK processed in the level of summarization.

USAGE NOTES:  If IMGWKSPK is smaller than WSQUOTA for the
user, all page faulting was a result of image activation.  If
IMGWKSPK is nearly as large as WSEXTENT, the process was
borrowing pages from free memory.  When IMGWKSPK is growing
to nearly the size of WSEXTENT, there is plenty of free
memory and the system is not swapping.  If paging is high and
this condition occurs, adjusting WSQUOTA and WSEXTENT higher
may improve overall performance.
```

B.2.4 - The VMS Accounting LOGIN Failed (DEA_LF) File

FILE ABSTRACT: VMS Accounting LOGIN Failed File	
CONTENT	Quantifies the resource usage of each LOGIN failure on the VMS system during the monitoring interval.
USE	To report resource usage of each LOGIN failure.
SOURCE	VMS ACCOUNTING record type 07 as documented in the VMS Accounting Reference Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	DAY	HOUR	ENDTS	
DAYS	N/A				
WEEKS	N/A				
MONTHS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	ZONE			
YEARS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	ZONE				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
X.....E	DAY	- Day of Month
X..MY.E	DEXACT1	- DIVISION
X..MY.E	DEXACT2	- UIC GROUP
X..MY.E	DEXACT3	- USER NAME
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..MY.E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year
X..MY.E	YEAR	- Year of Century
X..MY.E	ZONE	- Time Zone
Common Data Elements		
X.....E	DAYNAME	- Name of Day of Week
X.....E	DEXACCNT	- Account Name

```
 X.....E  DEXGRP   - User Identification Code Group
 X.....E  DEXJOB   - Jobname
 X.....E  DEXJOBID - Job ID
 X.....E  DEXMBR   - User Identification Code Member
 X.....E  DEXNADDR - Node Address
 X.....E  DEXNODE  - Nodename
 X.....E  DEXOWNID - Owner ID
 X.....E  DEXPID   - Process ID
 X.....E  DEXPRIV1 - Privilege 1 Mask
 X.....E  DEXPRIV2 - Privilege 2 Mask
 X.....E  DEXPRTY  - Priority
 X.....E  DEXQUEUE - Queue Name
 X.....E  DEXRMTID - Remote ID
 X.....E  DEXSTATS - Status
 X.....E  DEXTERM  - Terminal Name
 X.....E  DEXUSER  - Username
 X..MY.E  DURATION - Recording Interval Time
 X..MY.E  ENDTS    - End Time Stamp
 X.....E  ORGSYSID - Originating System Identification
 X..MY.E  STARTTS  - Start Time Stamp

Accumulated Data Elements

 X..MY.E  PRCBUFIO - Buffered I/Os
 X..MY.E  PRCCOST  - Processing Charges
 X..MY.E  PRCCPUNI - Number of Instructions Executed
 X..MY.E  PRCCPUTM - Total CPU Time
 X..MY.E  PRCDIRIO - Direct I/Os
 X..MY.   PRCFLTIO - Page Fault I/Os
 X..MY.   PRCIMAGE - Image Count
 X..MY.   PRCPAGFL - Page File Peak Usage
 X..MY.   PRCPAGFT - Page Faults
 X..MY.E  PRCVOLS  - Volumes Mounted

Maximum Data Elements

 X..MY.   PRCWKSPK - Working Set Peak
```

B.2.5 - The VMS Accounting Print Queued (DEAPRQ) File

FILE ABSTRACT: VMS Accounting Print Queued File	
CONTENT	Quantifies the resource usage of print queue on the VMS system during the monitoring interval.
USE	To report resource usage of each print queue.
SOURCE	VMS ACCOUNTING record type 08 as documented in the VMS Accounting Reference Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	DAY	HOUR	ENDTS	
DAYS	N/A				
WEEKS	N/A				
MONTHS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	ZONE			
YEARS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	ZONE				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
X.....E	DAY	- Day of Month
X..MY.E	DEXACT1	- DIVISION
X..MY.E	DEXACT2	- UIC GROUP
X..MY.E	DEXACT3	- USER NAME
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..MY.E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year
X..MY.E	YEAR	- Year of Century
X..MY.E	ZONE	- Time Zone
Common Data Elements		
X.....E	DAYNAME	- Name of Day of Week
X.....E	DEXACCNT	- Account Name

```
 X.....E  DEXGRP   - User Identification Code Group
 X.....E  DEXJOB   - Jobname
 X.....E  DEXJOBID - Job ID
 X.....E  DEXMBR   - User Identification Code Member
 X.....E  DEXNADDR - Node Address
 X.....E  DEXNODE  - Nodename
 X.....E  DEXOWNID - Owner ID
 X.....E  DEXPID   - Process ID
 X.....E  DEXPRIV1 - Privilege 1 Mask
 X.....E  DEXPRIV2 - Privilege 2 Mask
 X.....E  DEXPRTY  - Priority
 X.....E  DEXQUEUE - Queue Name
 X.....E  DEXRMTID - Remote ID
 X.....E  DEXSTATS - Status
 X.....E  DEXTERM  - Terminal Name
 X.....E  DEXUSER  - Username
 X..MY.E  DURATION - Recording Interval Time
 X..MY.E  ENDTS    - End Time Stamp
 X.....E  ORGSYSID - Originating System Identification
 X..MY.E  STARTTS  - Start Time Stamp

Accumulated Data Elements

 X..MY.E  PRQCOST  - Processing Charges
 X..MY.E  PRQGETS  - GET Count
 X..MY.E  PRQPAGES - Print Pages
 X..MY.E  PRQQIOS  - QIO Count
 X..MY.E  PRQQUETM - Queue Time
 X..MY.E  PRQSYMTM - Symbiont Time

Minimum Data Elements

 X..MY.E  PRQQUETS - Queue Time Stamp
```

```
+-----------------+
| P R Q C O S T   |  Processing Charges
+-----------------+
```

PRQCOST contains the total processing costs for a print job. The value of PRQCOST is zero unless a user computation is performed or the file containing this element is selected for billing processing by the CA MICS Accounting and Chargeback component.

TYPE: Computed data element - Numeric, measurement unit is real number.

SEE ALSO: IMGCOST - Processing Charges
PRCCOST - Processing Charges

ELEMENT DERIVATION: In the DETAIL timespan, PRQCOST is set by the CA MICS Accounting and Chargeback component interface.

RANGE OF VALUE: The value of this element is determined by the charging algorithms in effect at your installation.

SUMMARIZATION PROCESS: PRQCOST contains the sum of all occurrences of PRQCOST that were included in the level of summarization.

USAGE NOTES: None

```
+-----------------+
| P R Q G E T S   |  GET Count
+-----------------+
```

PRQGETS contains the number of RMS GET functions executed by the printer process to complete the job.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSSEQGT - Sequential GETs

ELEMENT DERIVATION: PRQGETS is set to the ACR$L_GETCNT field in the ACR$K_PRINT packet of the VMS ACCOUNTING PRINT record.

RANGE OF VALUE: PRQGETS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: PRQGETS contains the sum of all values of PRQGETS processed in the level of summarization.

USAGE NOTES: Printer files are RMS sequential files. The printer process (SYMBIONT) reads the file to be printed using the RMS GET operation. Because the file may have variable length records, the number of GETs may not related to the number of lines printed.

```
+-----------------+
| P R Q P A G E S |  Print Pages
+-----------------+
```

PRQPAGES contains the total number of pages printed by the printer process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: PRQPAGES is set to the value of the

ACR$L_PAGECNT field in the ACR$K_PRINT packet of the VMS ACCOUNTING PRINT record.

RANGE OF VALUE: PRQPAGES takes any integer value from 0 to 2,147,483,647

SUMMARIZATION PROCESS: PRQPAGES contains the sum of all values of PRQPAGES processed in the level of summarization.

USAGE NOTES: The printer process (SYMBIONT) keeps track of line counts to insert new pages when needed. It also intercepts and processes form control characters. The count of pages is a measure of resource cost for printing the job. See PRQSYMTM for another printing resource cost measure.

If the printer has been put into PASSALL mode by issuing a command to the queue manager, the SYMBIONT no longer can keep track of printer pages and form control characters. Line counts can be meaningless also. Under these conditions, PRQPAGES is not much use.

P R Q Q I O S | QIO Count

PRQQIOS contains the number of queued I/O requests sent by the printer process to complete the print job.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: PRQQIOS is set to the value of the ACR$L_QIOCNT field of the ACR$K_PRINT packet of the VMS ACCOUNTING PRINT record.

RANGE OF VALUE: PRQQIOS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: PRQQIOS contains the sum of all values of PRQQIOS processed in the level of summarization.

USAGE NOTES: QIOs are low-level routines that allow for relatively direct control of I/O devices. The printer process (SYMBIONT) uses QIOs to communicate with the printer when printing the job. QIOs can be of any size, so there is no good way of determining the job size from PRQQIOS, although the size of QIOs will be constant for a given application.

P R Q Q U E T M | Queue Time

PRQQUETM contains the amount of time the process was on the printer queue waiting to begin.

TYPE: Accumulated - Numeric, measurement unit is SAS time value.

SEE ALSO: Not applicable

ELEMENT DERIVATION: PRQQUETM is set to the value of the difference between the ACR$Q_BEGTIME and ACR$Q_QUETIME fields in the ACR$K_PRINT packet of the VMS ACCOUNTING PRINT record.

RANGE OF VALUE: PRQQUETM is a SAS time value ranging from 00.00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: PRQQUETM contains the sum of all values of PRQQUETM processed in the level of summarization.

USAGE NOTES: Large amounts of queue time can indicate printer backlog. A backlog of print jobs can be caused by jobs with a large number of lines to print or if the printer is inoperative (not ready, power off, malfunctioning, etc.). If the PRQQUETM is large and users are complaining about print turnaround time, the VAX system manager can enable job ordering by size or add more queues for the smaller jobs.

P R Q Q U E T S | Queue Time Stamp

The date and time the print job was queued.

TYPE: Minimum data element - SAS datetime value.

SEE ALSO: PRQQUETM - Queue Time

ELEMENT DERIVATION: PRQQUETS is set to the value of the ACR$Q_QUETIME field from the ACR$K_PRINT packet of the VMS ACCOUNTING PRINT QUEUED record.

RANGE OF VALUE: Any valid SAS time-stamp value. The value of this data element is not affected by your use of the 13MONTHYEAR option in prefix.MICS.PARMS(SITE).

SUMMARIZATION PROCESS: The minimum PRQQUETS encountered in summarizing a number of records to a higher level of granularity.

USAGE NOTES: None

| P R Q S Y M T M | Symbiont Time

PRQSYMTM contains the amount of time the CPU spent running the printer process (SYMBIONT) to print the job.

TYPE: Accumulated - Numeric, measurement unit is SAS time value.

SEE ALSO: Not applicable

ELEMENT DERIVATION: PRQSYMTM is set to the value of the ACR$L_SYMCPUTIM field in the ACR$K_PRINT packet of the VMS ACCOUNTING PRINT record.

RANGE OF VALUE: PRQSYMTM is a SAS time value ranging from 00.00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: PRQSYMTM contains the sum of all values of PRQSYMTM processed in the level of summarization.

USAGE NOTES: PRQSYMTM is a measure of resource cost for the printing the job. See PRQPAGES for another printer resource cost measurement.

FILE ABSTRACT: VMS Accounting Initialization File	
CONTENT	Quantifies the resource usage of each initialization during the monitoring interval.
USE	To report resource usage of each initialization.
SOURCE	VMS ACCOUNTING record type 05 as documented in the VMS Accounting Reference Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH	DAY	HOUR	ENDTS	
DAYS	N/A				
WEEKS	N/A				
MONTHS	N/A				
YEARS	N/A				
TABLES	N/A				

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
X.....E	DAY	- Day of Month
X.....E	DEXACT1	- DIVISION
X.....E	DEXACT2	- UIC GROUP
X.....E	DEXACT3	- USER NAME
X.....E	HOUR	- Hour of Day
X.....E	MONTH	- Month of Year
X.....E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year
X.....E	YEAR	- Year of Century
X.....E	ZONE	- Time Zone
Common Data Elements		
X.....E	DAYNAME	- Name of Day of Week
X.....E	DEXACCNT	- Account Name
X.....E	DEXGRP	- User Identification Code Group

```
 X.....E  DEXJOB   - Jobname
 X.....E  DEXJOBID - Job ID
 X.....E  DEXMBR   - User Identification Code Member
 X.....E  DEXNADDR - Node Address
 X.....E  DEXNODE  - Nodename
 X.....E  DEXOWNID - Owner ID
 X.....E  DEXPID   - Process ID
 X.....E  DEXPRIV1 - Privilege 1 Mask
 X.....E  DEXPRIV2 - Privilege 2 Mask
 X.....E  DEXPRTY  - Priority
 X.....E  DEXQUEUE - Queue Name
 X.....E  DEXRMTID - Remote ID
 X.....E  DEXSTATS - Status
 X.....E  DEXTERM  - Terminal Name
 X.....E  DEXUSER  - Username
 X.....E  DURATION - Recording Interval Time
 X.....E  ENDTS    - End Time Stamp
 X.....E  ORGSYSID - Originating System Identification
 X.....E  STARTTS  - Start Time Stamp
```

Accumulated Data Elements

```
 X.....E  INTBUFIO - Buffered I/Os
 X.....E  INTCPUNI - Number of Instructions Executed
 X.....E  INTCPUTM - CPU Time
 X.....E  INTDIRIO - Direct I/Os
 X.....   INTFLTIO - Page Fault I/Os
 X.....   INTIMAGE - Image Count
 X.....   INTPAGFL - Page File Peak Usage
 X.....   INTPAGFT - Page Faults
 X.....E  INTVOLS  - Volumes Mounted
```

Maximum Data Elements

```
 X.....   INTWKSPK - Working Set Peak
```

```
+-----------------+
| I N T B U F I O |  Buffered I/Os
+-----------------+
```

INTBUFIO contains the number of buffered I/Os requested by the initialization process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRCBUFIO - Buffered I/Os

ELEMENT DERIVATION: INTBUFIO is set to the value of the ACR$L_BIOCNT field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING INIT record.

RANGE OF VALUE: INTBUFIO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: INTBUFIO contains the sum of all values of INTBUFIO processed in the level of summarization.

USAGE NOTES: None

```
+-----------------+
| I N T C P U N I |  Number of Instructions Executed
+-----------------+
```

INTCPUNI contains an estimate of the total number of CPU instructions executed, calculated by multiplying the measured CPU time by a user-supplied VUPS (where VAX Units of Power for VAX 11/780=1) rate.

TYPE: Accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: CPUTM*MIPS

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: INTCPUNI contains the sum of all occurrences of INTCPUNI that were included in the level of summarization.

USAGE NOTES: INTCPUNI is an approximation based on system engineering concepts that were valid for second-generation equipment with hard-wired instructions and single-threaded execution. The accuracy of this element is a function of the instruction mix used in the program, the contending load (if any) using the system concurrently with the step, the processor architecture, the system microcode level, and the MIPS rate supplied by the user during the CA MICS generation process. Altering any of these factors will change the reported value of INTCPUNI for replicate runs, even though the exact code executes each time a step is initiated.

When used with caution, INTCPUNI can be helpful in comparing the relative efficiency of different software realizations of the same function (e.g., the relative path lengths of two database management systems for a backup/reload cycle performed on equivalent collections of data).

| I N T C P U T M | CPU Time

INTCPUTM contains the amount of time the CPU spent in the initialization process.

TYPE: Accumulated - Numeric, measurement unit is SAS time value, 4 bytes long.

SEE ALSO: PRCCPUTM - Total Process CPU Time

ELEMENT DERIVATION: INTCPUTM is set to the value of the ACR$L_CPUTM field in the ACR$K_RESOURCE packet of the VMS ACCOUNTING INIT record.

RANGE OF VALUE: INTCPUTM is a SAS time value ranging from 00:00.00 to the size of the DURATION measurement interval.

SUMMARIZATION PROCESS: INTCPUTM contains the sum of all values of INTCPUTM processed in the level of summarization.

USAGE NOTES: None

| I N T D I R I O | Direct I/Os

INTDIRIO contains the number of direct I/Os requested by the initialization process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: IMGDIRIO - Direct I/Os

ELEMENT DERIVATION: INTDIRIO is set to the value of the ACR$L_DIOCNT field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING INIT record.

RANGE OF VALUE: INTDIRIO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: INTDIRIO contains the sum of all values of INTDIRIO processed in the level of summarization.

USAGE NOTES: None

| I N T F L T I O | Page Fault I/Os

INTFLTIO contains the number of fault I/Os generated by the initialization process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRCFLTIO - Page Fault I/Os

ELEMENT DERIVATION: INTFLTIO is set to the value of the ACR$L_FAULTIO field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING INIT record.

RANGE OF VALUE: INTFLTIO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: INTFLTIO contains the sum of all values of INTFLTIO processed in the level of summarization.

USAGE NOTES: None

| I N T I M A G E | Image Count

INTIMAGE contains the number of images activated by the initialization process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRCIMAGE - Image Count

ELEMENT DERIVATION: INTIMAGE is set to the value of the ACR$L_IMGCNT field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING INIT record.

RANGE OF VALUE: INTIMAGE takes any integer value from 1 to 2,147,483,647.

SUMMARIZATION PROCESS: INTIMAGE contains the sum of all values of INTIMAGE processed in the level of summarization.

USAGE NOTES: None

I N T P A G F L Page File Peak Usage

INTPAGFL contains the maximum number of pages used by the initialization process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: IMGPAGFL - Page File Peak Usage

ELEMENT DERIVATION: INTPAGFL is set to the value of the ACR$L_PAGEFL field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING INIT record.

RANGE OF VALUE: INTPAGFL takes any integer value from 0 to the largest value of PGFLQUO for any user on the system.

SUMMARIZATION PROCESS: INTPAGFL contains the sum of all values of INTPAGFL processed in the level of summarization.

USAGE NOTES: INTPAGFL is useful if you want to know the size of the process that initialized the VMS accounting file.

I N T P A G F T Page Faults

INTPAGFT contains the number of page faults generated by the initialization process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: IMGPAGFT - Page Faults
INTFLTIO - Page Fault I/Os

ELEMENT DERIVATION: INTPAGFT is set to the value of the ACR$L_FAULTS field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING INIT record.

RANGE OF VALUE: INTPAGFT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: INTPAGFT contains the sum of all values of INTPAGFT processed in the level of summarization.

USAGE NOTES: None

I N T V O L S Volumes Mounted

INTVOLS contains the number of volume mounts requested by the initialization process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: IMGVOLS - Volumes Mounted

ELEMENT DERIVATION: INTVOLS is set to the value of the ACR$L_VOLUMES field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING INIT record.

RANGE OF VALUE: INTVOLS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: INTVOLS contains the sum of all values of INTVOLS processed in the level of summarization.

USAGE NOTES: INTVOLS should always be 0, because there is no mount associated with the VMS accounting.

I N T W K S P K Working Set Peak

INTWKSPK contains the maximum working set size of the initialization process.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: IMGWKSPK - Working Set Peak

ELEMENT DERIVATION: INTWKSPK is set to the value of the ACR$L_WSPEAK field in the ACR$K_RESOURCES packet of the VMS ACCOUNTING INIT record.

RANGE OF VALUE: INTWKSPK ranges from an integer value 1 to the smallest value specified by the:
- o SYSGEN parameter WSMAX, or
- o User authorization file parameter WSEXTENT

SUMMARIZATION PROCESS: INTWKSPK contains the sum of all values of INTWKSPK processed in the level of summarization.

USAGE NOTES: INTWKSPK is useful if you want to know the size of the working set for the initialization process.

B.3 - VAX/VMS Monitor Information (DEM) Area

The sections that follow describe the VAX/VMS Monitor information area (DEM). The statistics for these files are obtained from the VMS MONITOR data.

1 - File Relationships and Common Data Elements
2 - The VMS Monitor Process (DEMPRO) File
3 - The VMS Monitor Modes (DEMMOD) File
4 - The VMS Monitor Disk (DEMDSK) File
5 - The VMS Monitor SCS (DEMSCS) File
6 - The VMS Monitor RMS (DEMRMS) File
7 - The VMS System Profile (DEMSPR) File

B.3.1 - File Relationships and Common Data Elements

Not available at this time.

B.3.2 - The VMS Monitor Process (DEMPRO) File

FILE ABSTRACT: VMS Monitor Process File	
CONTENT	Quantifies the resource usage of each process during the monitoring interval.
USE	To report resource usage of each process for performance tuning and capacity planning.
SOURCE	VMS MONITOR record type 0 as documented in the VMS Monitor Reference Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	PROPID	PROPNAME	YEAR	MONTH
	DAY	HOUR	ENDTS		
DAYS	SYSID	PROPNAME	YEAR	MONTH	DAY
	HOUR				
WEEKS	SYSID	PROPNAME	YEAR	WEEK	ZONE
	HOUR				
MONTHS	SYSID	PROPNAME	YEAR	MONTH	ZONE
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year
X.....E	PROPID	- Process ID
XDWM..E	PROPNAME	- Process Name
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
XDWM..E	ZONE	- Time Zone
Common Data Elements		
XD....E	DAYNAME	- Name of Day of Week
XDWM..	DEXBOOTS	- Monitor System Boot Time Stamp
XDWM..E	DEXDUR	- Requested Recording Interval Time

```
 X.....E  DEXINTNO - Reserved Field Incremented by Monitor
 X.....E  DEXMONVR - VMS Monitor Version
 XDWM..E  DEXTS    - Monitor Time Stamp
 XDWM..E  DURATION - Recording Interval Time
 XDWM..E  ENDTS    - End Time Stamp
 XDWM..E  INTERVLS - Number of Recording Intervals
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  STARTTS  - Start Time Stamp
```

Retained Data Elements

```
 X.....E  PROEID   - Extended Process ID
 X.....   PROEVNTF - Event Flag Wait Mask
 X.....E  PROGLBPG - Global Page Count
 X.....E  PROGRP   - User Identification Code Group
 X.....E  PROMBR   - User Identification Code Member
 X.....E  PROPRCPG - Process Page Count
 X.....   PROPRTY  - Priority
 X.....E  PROSTATE - State
 X.....   PROSTATF - Status Flags
```

Accumulated Data Elements

```
 XDWM..E  PROBUFIO - Buffered I/Os
 XDWM..E  PROCPUNI - Number of Instructions Executed
 XDWM..E  PROCPUTM - CPU Time
 XDWM..E  PRODIRIO - Direct I/Os
 XDWM..E  PROPGFLT - Page Faults
```

Maximum Data Elements

```
 XDWM..E  PROMXGPG - Maximum Global Page Count
 XDWM..E  PROMXPPG - Maximum Process Page Count
```

P R O B U F I O Buffered I/Os

PROBUFIO contains the count of I/O requests to buffered devices, such as terminals and printers.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO:
- SPRBIOS - Buffered I/Os
- PRCBUFIO - Buffered I/Os
- IMGBUFIO - Buffered I/Os
- INTBUFIO - Buffered I/Os

ELEMENT DERIVATION: PROBUFIO is set to value of the MNR_PRO$L_BIOCNT field of the VMS MONITOR PROCESS record.

RANGE OF VALUE: PROBUFIO takes any numeric value in the range from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: PROBUFIO contains the sum of all values of PROBUFIO processed in the level of summarization.

USAGE NOTES: PROBUFIO reflects the number of buffered I/O operations, not the size of those operations.

P R O C P U N I Number of Instructions Executed

PROCPUNI contains an estimate of the total number of CPU instructions executed for a process, calculated by multiplying the measured CPU time by a user-supplied VUPS (where VAX Units of Power for VAX 11/780=1) rate.

TYPE: Accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: PROCPUTM*MIPS

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: PROCPUNI contains the sum of all occurrences of PROCPUNI that were included in the level of summarization.

USAGE NOTES: PROCPUNI is an approximation based on system engineering concepts that were valid for second-generation equipment with hard-wired instructions and single-threaded execution. The accuracy of this element is a function of the instruction mix used in the program, the contending load (if any) using the system concurrently with the step, the processor architecture, the system microcode level, and the MIPS rate supplied by the user during the CA MICS generation process. Altering any of these factors will change the reported value of PROCPUNI for replicate runs, even though the exact code executes each time a step is initiated.

When used with caution, PROCPUNI can be helpful in comparing the relative efficiency of different software realizations of the same function (e.g., the relative path lengths of two database management systems for a backup/reload cycle performed on equivalent collections of data).

```
+-----------------+
| P R O C P U T M |  CPU Time
+-----------------+
PROCPUTM contains the amount of CPU time used by the process
for DURATION (measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is SAS time
value.

SEE ALSO:  PRCCPUTM - Total Process CPU Time
           IMGCPUTM - Total Image CPU Time
           INTCPUTM - CPU Time

ELEMENT DERIVATION: PROCPUTM is set to value of the
MNR_PRO$L_CPUTIM field of the VMS MONITOR PROCESS record.

RANGE OF VALUE:  PROCPUTM is a SAS time value in the range
from 0 to the size of the DURATION (measurement interval)
value.

SUMMARIZATION PROCESS:  PROCPUTM contains the sum of all
values of PROCPUTM processed in the level of summarization.

USAGE NOTES:  When the process is swapped out, this value is
set to zero.  PROCPUTM includes time spent only in user and
supervisor modes.  If the system spends CPU time in kernel or
executive mode on behalf of a process, but not under the
process' direct control, the CPU time is not included in this
field.



+-----------------+
| P R O D I R I O |  Direct I/Os
+-----------------+
PRODIRIO contains the count of I/O requests to nonbuffered
devices, such as disks or tape drives.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO:  SPRDIOS  - Direct I/Os
           PRCDIRIO - Direct I/Os
           IMGDIRIO - Direct I/Os
           INTDIRIO - Direct I/Os

ELEMENT DERIVATION: PRODIRIO is set to value of the
MNR_PRO$L_DIOCNT field of the VMS MONITOR PROCESS record.

RANGE OF VALUE:  PRODIRIO takes any numeric value in the
range from 0 to 2,147,483,647.

SUMMARIZATION PROCESS:  PRODIRIO contains the sum of all
values of PRODIRIO processed in the level of summarization.

USAGE NOTES:  None



+-----------------+
| P R O E I D     |  Extended Process ID
+-----------------+
PROEID contains the extended ID number assigned by VMS when
the process is created.

TYPE:  Retained - Numeric, measurement unit is hexadecimal
integer, 4 bytes long.

SEE ALSO:  DEXPID - Process ID

ELEMENT DERIVATION: PROEID is set to the value of the
MNR_PRO$L_EPID field of the VMS MONITOR PROCESS record.

RANGE OF VALUE:  PROEID takes any hexadecimal numeric value
in the range from 00000001 to FFFFFFFF.

SUMMARIZATION PROCESS:  PROEID contains the last value of
PROEID processed in the level of summarization.

USAGE NOTES:  The PROEID number in the DETAIL timespan of
the VMS MONITOR Process files can be used to coordinate
information between VMS MONITOR data and that obtained from
VMS ACCOUNTING.  The VMS ACCOUNTING record shows the total
usage of resources by a process, and the MONITOR data shows
when those resources were used.  If image accounting data is
being captured, the resources used by a process can be traced
to the images in use when that process was using resources
abnormally.  If only process accounting data is being kept,
PROPID can be used to determine when and how much resources a
process used at smaller granularities (DURATION (measurement
interval)).

The DEXPID number in the DETAIL timespan of the CA MICS VMS
ACCOUNTING files can be used to correlate data between VMS
MONITOR and VMS ACCOUNTING.  PROEID can be used to find when
a process used most of its resources.  VMS ACCOUNTING shows a
total usage of resources by a process and VMS MONITOR shows
when those resources were broken down by DURATION
(measurement interval).
```

| P R O E V N T F | Event Flag Wait Mask

PROEVNTF contains the event flag wait bit mask of the process.

TYPE: Retained - Bit mask, 32 bits.

SEE ALSO: Not applicable

ELEMENT DERIVATION: PROEVNTF is set to the value of the MNR_PRO$L_EFWM field of the VMS MONITOR PROCESS record.

RANGE OF VALUE: PROEVNTF takes any hexadecimal value from 0 to FFFFFFFF.

SUMMARIZATION PROCESS: PROEVNTF contains the last value of PROEVNTF processed in the level of summarization.

USAGE NOTES: None

| P R O G L B P G | Global Page Count

PROGLBPG contains the count of global pages mapped to this process.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROMXGPG - Maximum Global Page Count

ELEMENT DERIVATION: PROGLBPG is set to the value of the MNR_PRO$L_GPGCNT field of the VMS MONITOR PROCESS record.

RANGE OF VALUE: PROGLBPG takes any numeric value from 0 to the number of global pages defined in VMS SYSGEN parameters.

SUMMARIZATION PROCESS: PROGLBPG contains the last value of PROGLBPG processed in the level of summarization.

USAGE NOTES: PROGLBPG indicates how much shared memory the process is using. A high value in PROGLBPG usually indicates a large amount of interprocess communication. Use of shared memory requires synchronization among processes to serialize access to the memory.

| P R O G R P | User Identification Code Group

PROGRP contains the high-order word of the user identification code for the owner of the process. These values are assigned by the VAX system manager.

TYPE: Retained - Numeric, measurement unit is octal integer.

SEE ALSO: PROMBR - User Identification Code Member
DEXGRP - User Identification Code Group

ELEMENT DERIVATION: PROGRP is set to the high-order word of the MNR_PRO$L_UIC field of the VMS MONITOR PROCESS record.

RANGE OF VALUE: PROGRP takes an octal integer value in the range 1 to 77777.

SUMMARIZATION PROCESS: PROGRP contains the last value of PROGRP processed in the level of summarization.

USAGE NOTES: PROGRP can be used to summarize detail records by group number. This summarization can show whether certain groups of users use resources at different levels from other groups. By itself, this value represents a class of VMS users; when combined with PROMBR, it is a unique identifier for an authorized VMS account and makes up the user identification code. If the value is between 1 and 37776, PROGRP indicates that the process is related to a user account. If the value is between 37777 and 77777, PROGRP indicates that a resource identifier (assigned by the VAX system manager) owns the process.

| P R O M B R | User Identification Code Member

PROMBR contains the low-order word of the user identification code for the owner of the process. These values are assigned by the VAX system manager.

TYPE: Retained - Numeric, measurement unit is octal integer.

SEE ALSO: PROGRP - User Identification Code Group
DEXMBR - User Identification Code Member

ELEMENT DERIVATION: PROMBR is set to the low-order word of the MNR_PRO$L_UIC field of the VMS MONITOR PROCESS record.

RANGE OF VALUE: PROMBR takes an octal integer value in the range 1 to 177777.

SUMMARIZATION PROCESS: PROMBR contains the last value of PROMBR processed in the level of summarization.

USAGE NOTES: PROMBR should be used only in conjunction with PROGRP to identify each unique VMS user. When combined with PROGRP, this is a unique identifier for authorized user accounts and makes up the user identification code. If the value is 177777, PROMBR indicates that the owner of the process is a resource identifier rather than a user account.

P R O M X G P G Maximum Global Page Count

PROMXGPG contains the maximum number of global pages used by the process.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROGLBPG - Global Page Count

ELEMENT DERIVATION: PROMXGPG is set to the maximum value of PROGLBPG reached.

RANGE OF VALUE: PROMXGPG takes any integer value from 0 to the number of global pages defined in VMS SYSGEN parameters.

SUMMARIZATION PROCESS: PROMXGPG contains the maximum value of PROMXGPG processed for the level of summarization.

USAGE NOTES: PROMXGPG indicates the maximum shared memory the process used. A high value in PROMXGPG usually indicates a large amount of interprocess communication. Use of shared memory requires synchronization among processes to serialize access to the memory.

P R O M X P P G Maximum Process Page Count

PROMXPPG contains the maximum number of local pages of memory used by the process.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROPRCPG - Process Page Count

ELEMENT DERIVATION: PROMXPPG is set to the maximum value reached by PROPRCPG.

RANGE OF VALUE: PROMXPPG takes any integer value from 0 to the page file quota set in the user account.

SUMMARIZATION PROCESS: PROMXPPG contains the maximum value of PROMXPPG processed for the level of summarization.

USAGE NOTES: PROMXPPG is 0 for the SWAPPER process. If PROMXPPG is larger than the user's working set size, or if it is larger than the working set quota and the system has run out of free pages, the process is subject to page faulting.

P R O P G F L T Page Faults

PROPGFLT contains the count of page faults for the duration of the interval.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRCPAGFT - Page Faults
IMGPAGFT - Page Faults
SPRPGEFL - PGE Page Faults

ELEMENT DERIVATION: PROPGFLT is set to the value of the MNR_PRO$L_PAGEFLTS field of the VMS MONITOR PROCESS record.

RANGE OF VALUE: PROPGFLT takes any integer value from 0 to 4,294,967,295.

SUMMARIZATION PROCESS: PROPGFLT contains the sum of all values of PROPGFLT processed in the level of summarization.

USAGE NOTES: A high value for PROPGFLT means that the process is not finding the pages of process data or program memory resident in the physical memory of the machine when it needs them. PROPGFLT is high for two primary reasons:

- o The process is continually activating new images.
- o The images are larger than the allowed working set.

Large amounts of page faulting will degrade system performance. If the size of physical memory permits, some page faulting can be reduced by increasing the working set size of the processes that tend to have high values in PROPGFLT.

| P R O P I D | Process ID

PROPID contains the internal process ID number assigned by VMS during process creation.

TYPE: Retained - Numeric, hexadecimal number unique on a system at any given time.

SEE ALSO: PROEID - Extended Process ID

ELEMENT DERIVATION: PROPID is set to the value of the MNR_PRO$L_PID field of the VMS MONITOR PROCESS record.

RANGE OF VALUE: PROPID takes any hexadecimal numeric value in the range from 00000001 to FFFFFFFF.

SUMMARIZATION PROCESS: PROPID contains the last value of PROPID processed in the level of summarization.

USAGE NOTES: The DEXPID number in the DETAIL time-span of the VMS ACCOUNTING files can be used to coordinate information with the VMS MONITOR data element PROEID. The PROPID number recorded in PROPID is local to VMS MONITOR PROCESSES record and is not used anywhere else.

| P R O P N A M E | Process Name

PROPNAME contains the name of the process.

TYPE: Retained - Character, length 15.

SEE ALSO: Not applicable

ELEMENT DERIVATION: PROPNAME is set to the value of the MNR_PRO$T_LNAME field of the VMS MONITOR PROCESS record. Source data contains counted character string, 1 byte per count, 15 bytes of characters.

RANGE OF VALUE: PROPNAME can contain any combination of 1 to 15 characters. The count byte from source data is not stored.

SUMMARIZATION PROCESS: PROPNAME is not summarized.

USAGE NOTES: None

| P R O P R C P G | Process Page Count

PROPRCPG contains the number of local pages of memory in use by the process.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROMXPPG - Maximum Process Page Count

ELEMENT DERIVATION: PROPRCPG is set to the value of the MNR_PRO$L_PPGCNT field of the VMS MONITOR PROCESS record.

RANGE OF VALUE: PROPRCPG takes any integer value from 0 to the page file quota set in the user account.

SUMMARIZATION PROCESS: PROPRCPG contains the last value of PROPRCPG processed in the level of summarization.

USAGE NOTES: PROPRCPG is 0 for the SWAPPER process. If PROPRCPG is larger than the user's working set size, or if it is larger than the working set quota and the system has run out of free pages, the process will begin page faulting. Processes with a large PROPRCPG are the first targets for swapping.

If the sum of PROPRCPG for all processes at a given time exceeds the size of the page file, VMS will stop the process that attempted to exceed the page file size. If the page file is fragmented, VMS may halt the system.

| P R O P R T Y | Priority

PROPRTY contains the priority level code for the process.

TYPE: Retained - Numeric, measurement unit is integer, 1 byte long.

SEE ALSO: DEXPRTY - Priority

ELEMENT DERIVATION: PROPRTY is set to the value of the field MNR_PRO$B_PRI from the VMS MONITOR PROCESS record.

RANGE OF VALUE: PROPRTY can be any integer value from 0 to 30.

SUMMARIZATION PROCESS: PROPRTY contains the last value of PROPRTY processed in the level of summarization.

USAGE NOTES: PROPRTY is the actual priority level subtracted from 31. If PROPRTY is 15 or less, the process is running in real-time mode and cannot be preempted by the scheduler. On most systems, SWAPPER is the only process running at that high a priority; its PROPRTY is 15. Larger values of PROPRTY mean lower priority for a process.

For processes with PROPRTY 16 or greater, the scheduler decreases PROPRTY each time it cannot schedule the process to run. When a process runs the scheduler, it sets PROPRTY to the base level for that process. The scheduler will never cause PROPRTY to become less than 16.

P R O S T A T E | State

PROSTATE contains a code for the processing state of the process.

TYPE: Retained - Numeric, measurement unit is integer, 2 bytes long.

SEE ALSO:
- SPRCOLPG - Collided Page Wait
- SPRMWAIT - Miscellaneous Resource Wait
- SPRCEF - Common Event Flag Wait
- SPRPFW - Page Fault Wait
- SPRLEF - Local Event Flag Wait, Inswapped
- SPRLEFO - Local Event Flag Wait, Outswapped
- SPRHIB - Hibernate Wait, Inswapped
- SPRHIBO - Hibernate Wait, Outswapped
- SPRSUSP - Suspended Wait, Inswapped
- SPRSUSPO - Suspended Wait, Outswapped
- SPRFPGWT - Free Page Wait
- SPRCOM - Compute State Wait, Inswapped
- SPRCOMO - Compute State Wait, Outswapped
- SPRCUR - Current Processes

ELEMENT DERIVATION: PROSTATE is set to the value of the MNR_PRO$W_STATE field of the VMS MONITOR PROCESS record.

RANGE OF VALUE: PROSTATE takes any integer value 1 to 14.

SUMMARIZATION PROCESS: PROSTATE contains the last value of PROSTATE processed in the level of summarization.

USAGE NOTES: PROSTATE is currently defined for 14 values, although DEC may define more values in the future. Each value and its meaning follow:

1 - COLPG collided page wait
2 - MWAIT miscellaneous resource wait
3 - CEF common event flag wait
4 - PFW page fault wait
5 - LEF local event flag wait, inswapped
6 - LEFO local event flag wait, outswapped
7 - HIB hibernate wait, inswapped
8 - HIBO hibernate wait, outswapped
9 - SUSP suspended wait, inswapped
10 - SUSPO suspended wait, outswapped
11 - FPGWT free page wait
12 - COM compute state wait, inswapped
13 - COMO compute state wait, outswapped
14 - CUR current processes

If a process is in either the HIB or HIBO state, it is waiting for some time to pass, and it is not affecting system performance nor is system activity affecting it. If a process is in either the LEF or LEFO state, it is probably waiting for user input and is also not affecting the system negatively. If a process is in either the SUSP or SUSPO state, another process has suspended its operation. A process in the COLPG, MWAIT, CEF, PFW, or FPGWT state is waiting for some other process to free some resource(s) before continuing.

Processes in LEFO, HIBO, SUSPO, and COMO states indicate that the system probably does not have enough memory for the work that it is doing. Processes in the COM and COMO states indicate that the CPU is not fast enough to keep up with the workload. Many processes in the COLPG, PFW, and FPGWT states indicate that working sets may not be large enough.

P R O S T A T F | Status Flags

PROSTATF contains the status flag indicators for the process.

TYPE: Retained - Numeric, measurement unit is hexadecimal integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: PROSTATF is set to the value of the MNR_PRO$L_STS field of the VMS MONITOR PROCESS record.

RANGE OF VALUE: PROSTATF takes any hexadecimal integer value in the range from 00000000 to FFFFFFFF.

SUMMARIZATION PROCESS: PROSTATF contains the last value of PROSTATF processed in the level of summarization.

USAGE NOTES: Bit 0 (PCB$V_RES) of PROSTATF is set to 1 when the process is in memory and 0 when it is swapped out. The recording intervals possible through VMS MONITOR, however, are not small enough to make PROSTATF very useful to VAX system managers.

B.3.3 - The VMS Monitor Modes (DEMMOD) File

FILE ABSTRACT: VMS Monitor Modes File	
CONTENT	Quantifies the time spent in each of the VAX processor modes.
USE	To report on the VAX processor modes for performance tuning and capacity planning.
SOURCE	VMS MONITOR record type 02 as documented in the VMS Monitor Reference Manual, plus the VMS Monitor record type 23 for MODVECTM element.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	MODCPU	YEAR	MONTH	DAY
	HOUR	ENDTS			
DAYS	SYSID	MODCPU	YEAR	MONTH	DAY
	HOUR				
WEEKS	SYSID	MODCPU	YEAR	WEEK	ZONE
	HOUR				
MONTHS	SYSID	MODCPU	YEAR	MONTH	ZONE
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
		Sequence/Summary Data Elements
XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XDWM..E	MODCPU	- CPU Identifier
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
XDWM..E	ZONE	- Time Zone
		Common Data Elements
XD....E	DAYNAME	- Name of Day of Week
XDWM..	DEXBOOTS	- Monitor System Boot Time Stamp
XDWM..E	DEXDUR	- Requested Recording Interval Time
X.....E	DEXINTNO	- Reserved Field Incremented by Monitor

```
 X.....E  DEXMONVR - VMS Monitor Version
 XDWM..E  DEXTS    - Monitor Time Stamp
 XDWM..E  DURATION - Recording Interval Time
 XDWM..E  ENDTS    - End Time Stamp
 XDWM..E  INTERVLS - Number of Recording Intervals
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  STARTTS  - Start Time Stamp
```

Accumulated Data Elements

```
 XDWM..E  MODCOMTM - Compatibility Mode Time
 XDWM..   MODCPUNI - Number of Instructions Executed
 XDWM..E  MODEXETM - Executive Mode Time
 XDWM..E  MODIDLTM - Idle Time
 XDWM..E  MODINTTM - Interrupt Stack Time
 XDWM..E  MODKERTM - Kernel Mode Time
 XDWM..E  MODMPSTM - Multi-Processor Sync Time
 XDWM..E  MODSUPTM - Supervisor Mode Time
 XDWM..E  MODUSRTM - User Mode Time
 XDWM..E  MODVECTM - CPU Time of Vector Consumers
```

Derived Data Elements

```
 XDWM..E  MODCPUTM - Total CPU Time
 XDWM..E  MODPCCOM - Compatibility Mode Percent
 XDWM..E  MODPCCPU - Total CPU Percent
 XDWM..E  MODPCEXE - Executive Mode Percent
 XDWM..E  MODPCIDL - Idle Percent
 XDWM..E  MODPCINT - Interrupt Stack Percent
 XDWM..E  MODPCKER - Kernel Mode Percent
 XDWM..E  MODPCMPS - Multi-Processor Sync Percent
 XDWM..E  MODPCSUP - Supervisor Mode Percent
 XDWM..E  MODPCUSR - User Mode Percent
```

M O D C O M T M Compatibility Mode Time

MODCOMTM contains the number of seconds the CPU spent in compatibility mode during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is SAS time value.

SEE ALSO: MODCPUTM - Total CPU Time

ELEMENT DERIVATION: MODCOMTM is set to the value of the MNR_MOD$L_COMPAT field of the VAX MONITOR MODES record minus the previous value.

RANGE OF VALUE: MODCOMTM is a SAS time value ranging from 00:00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: MODCOMTM contains the sum of all values of MODCOMTM processed in the level of summarization.

USAGE NOTES: MODCOMTM measures compatibility mode, that is, the time spent by the VAX CPU emulating PDP-11 instructions. Since DEC no longer makes VAX processors that run these instructions, the value is generally 0. The VAX models that support this mode are the 11/7xx and 8xxx lines. Compatibility mode will severely affect performance if true VAX processing must occur at the same time.

M O D C P U CPU Identifier

MODCPU contains an identifier of the CPU associated with the VMS MONITOR MODES record and the MONITOR VECTOR record.

TYPE: Retained - Numeric, measurement unit is integer, 1 byte long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: MODCPU is set to the value of the MNR_MOD$B_CPUID field of the VMS MONITOR MODES record. When merging VECTOR record information with the MODES record, the MNR_VEC$B_CPUID field of the VMS MONITOR VECTOR record is used as the CPU identifier portion of the merge key.

RANGE OF VALUE: MODCPU can be any integer value from 1 to the number of processors installed on the VAX hardware.

SUMMARIZATION PROCESS: MODCPU is not summarized. MODCPU is used to determine a unique key for records within the Modes file based on the time-span so other elements in the file can be summarized.

USAGE NOTES: MODCPU is a field that the VMS MONITOR MODES record uses to report system usage by processor in multiprocessor VAX machines. In single processor systems, MODCPU is always 1.

M O D C P U N I Number of Instructions Executed

MODCPUNI contains an estimate of the total number of CPU instructions executed, calculated by multiplying the measured CPU time by a user-supplied VUPS (where VAX Units of Power for VAX 11/780=1) rate.

TYPE: Accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: MODCPUTM*MIPS

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: MODCPUNI contains the sum of all occurrences of MODCPUNI that were included in the level of summarization.

USAGE NOTES: MODCPUNI is an approximation based on system engineering concepts that were valid for second-generation equipment with hard-wired instructions and single-threaded execution. The accuracy of this element is a function of the instruction mix used in the program, the contending load (if any) using the system concurrently with the step, the processor architecture, the system microcode level, and the MIPS rate supplied by the user during the CA MICS generation process. Altering any of these factors will change the reported value of MODCPUNI for replicate runs, even though the exact code executes each time a step is initiated.

When used with caution, MODCPUNI can be helpful in comparing the relative efficiency of different software realizations of the same function (e.g., the relative path lengths of two database management systems for a backup/reload cycle performed on equivalent collections of data).

M O D C P U T M Total CPU Time

MODCPUTM contains the total number of seconds in which the CPU was being used.

TYPE: Computed using specific expression statements - Numeric, measurement unit is SAS time value.

SEE ALSO: MODCOMTM - Compatibility Mode Time
MODEXETM - Executive Mode Time
MODINTTM - Interrupt Stack Time
MODKERTM - Kernel Mode Time
MODMPSTM - Multi-Processor Sync Time
MODSUPTM - Supervisor Mode Time
MODUSRTM - User Mode Time
DURATION - Length of Recording Interval

ELEMENT DERIVATION: MODCPUTM is computed using the following algorithm:

```
MODCPUTM = SUM (MODCOMTM, MODEXETM, MODINTTM, MODKERTM,
                MODMPSTM, MODSUPTM, MODUSRTM)
```

RANGE OF VALUE: MODCPUTM is a SAS time value ranging from 00:00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: MODCPUTM contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: MODCPUTM collects all the time the VAX CPU spent doing user or overhead processing. If the value is equal or close to DURATION (measurement interval) then performance is probably bad because of a CPU that is not fast enough for the workload.

M O D E X E T M Executive Mode Time

MODEXETM contains the number of seconds spent in executive mode through DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is SAS time value.

SEE ALSO: MODCPUTM - Total CPU Time

ELEMENT DERIVATION: MODEXETM is set to the incremental value of the MNR_MOD$L_EXEC field of the VMS MONITOR MODES record.

RANGE OF VALUE: MODEXETM is a SAS time value ranging from 00:00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: MODEXETM contains the sum of all values of MODEXETM processed in the level of summarization.

USAGE NOTES: MODEXETM represents the time the CPU is using in executive mode, usually Record Management System (RMS) activity. If the value is high, the RMS subsystem should be tuned by finding more appropriate values for index buffering, bucket sizes, fill factors, and use of alternate keys.

M O D I D L T M | Idle Time

MODIDLTM contains the number of seconds the CPU spent in idle mode during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is SAS time value.

SEE ALSO: MODCPUTM - Total CPU Time

ELEMENT DERIVATION: MODIDLTM is set to the value of the MNR_MOD$L_IDLE field from the VMS MONITOR MODES record minus the previous value.

RANGE OF VALUE: MODIDLTM is a SAS time value ranging from 00:00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: MODIDLTM contains the sum of all values of MODIDLTM processed in the level of summarization.

USAGE NOTES: MODIDLTM is not just time that the CPU is not busy. A very busy system could still have a relatively large MODIDLTM if the processes were all being blocked by the lack of availability of other resources. If system performance is poor and idle time is a large amount, start investigating the I/O subsystem, the network activity, and the locking of resources between processors.

M O D I N T T M | Interrupt Stack Time

MODINTTM contains the number of seconds the CPU spent in interrupt stack mode during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is SAS time value.

SEE ALSO: MODCPUTM - Total CPU Time

ELEMENT DERIVATION: MODINTTM is set to the value of the MNR_MOD$L_INTER field of the VAX MONITOR MODES record minus the previous value.

RANGE OF VALUE: MODINTTM is a SAS time value ranging from 00:00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: MODINTTM contains the sum of all values of MODINTTM processed in the level of summarization.

USAGE NOTES: On a nonclustered system, MODINTTM usually indicates a high number of interrupts from peripheral devices such as disks, tapes, printers, and terminals. If MODINTTM is large and it seems that terminals and printers may be at fault, consider changing the configuration to use terminal servers for those devices.

On a clustered system, MODINTTM may represent time spent serving distributed locks or providing disk and tape services to other members of the cluster. If MODINTTM is large on a clustered system, monitor the DLOCK, SCS, and MSCP values.

M O D K E R T M | Kernel Mode Time

MODKERTM contains the number of seconds the CPU spent in kernel mode during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is SAS time value.

SEE ALSO: MODCPUTM - Total CPU Time

ELEMENT DERIVATION: MODKERTM is set to the value of the MNR_MOD$L_KERNEL field of the VAX MONITOR MODES record minus the previous value.

RANGE OF VALUE: MODKERTM is a SAS time value ranging from 00:00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: MODKERTM contains the sum of all values of MODKERTM processed in the level of summarization.

USAGE NOTES: A high value in MODKERTM (25 percent or more of MODCPUTM) indicates an overuse of some system-wide resources:

- o Memory limitations could be causing paging and swapping.
- o Excessive locking will use global sections and require the interaction of the VMS Lock Manager.
- o Excessive process creation requires allocation of control blocks as well as high amounts of paging.

- o Excessive image activation can cause high page faulting and logical name translation.
- o Excessive file system activity such as file creation, opens and closes, file extending, and file deletions requires kernel mode activities.
- o Excessive DECnet traffic requires kernel mode intervention, even if the packets are not addressed to this node.

M O D M P S T M | Multi-Processor Sync Time

MODMPSTM contains the number of seconds the CPU spent in multiprocessor sync mode synchronizing operations between processors during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is SAS time value.

SEE ALSO: MODCPUTM - Total CPU Time

ELEMENT DERIVATION: MODMPSTM is set to the value of the MNR_MOD$L_MPSYNC field of the VAX MONITOR MODES record minus the previous value.

RANGE OF VALUE: MODMPSTM is a SAS time value ranging from 00:00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: MODMPSTM contains the sum of all values of MODMPSTM processed in the level of summarization.

USAGE NOTES: MODMPSTM is a good indicator of how much time a multiprocessor VAX system is spending trying to coordinate access to system resources used by the different processors. If MODMPSTM is 8 percent or more of MODCPUTM, further investigation of paging, I/O, and lock management is warranted.

M O D P C C O M | Compatibility Mode Percent

MODPCCOM contains the percentage of time the CPU spent in compatibility mode during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: MODCOMTM - Compatibility Mode Time

ELEMENT DERIVATION: MODPCCOM is computed using the following algorithm:

```
if DURATION > 0 then
  MODPCCOM = MODCOMTM * 100 / DURATION;
else
  MODPCCOM = .;
```

RANGE OF VALUE: The value of MODPCCOM ranges from 0 to 100.

SUMMARIZATION PROCESS: MODPCCOM contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: MODPCCOM measures the percentage of time the CPU spends in compatibility mode emulating PDP-11 instructions. Because DEC no longer makes VAX processors that run these instructions, the value is generally 0. The VAX models that support this mode are the 11/7xx and 8xxx lines. Operating in compatibility mode will severely degrade performance if true VAX processing must occur at the same time.

M O D P C C P U | Total CPU Percent

MODPCCPU contains the percentage of time spent in total CPU mode during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: MODCPUTM - Total CPU Time
MODPCINT - Interrupt Stack Percent
MODPCKER - Kernel Mode Percent
MODPCEXE - Executive Mode Percent
MODPCIDL - Idle Percent

ELEMENT DERIVATION: MODPCCPU is computed using the following algorithm:

```
if DURATION > 0 then
  MODPCCPU = MODCPUTM * 100 / DURATION;
else
  MODPCCPU = .;
```

RANGE OF VALUE: The value of MODPCCPU ranges from 0 to 100.

SUMMARIZATION PROCESS: MODPCCPU contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: MODPCCPU should be close to 100 minus MODPCIDL. MODPCCPU is very important in determining the balance between the modes because it is percent of busy time that makes the difference. In general, the sum of MODPCINT, MODPCKER, and MODPCEXE should not exceed 40 percent of MODPCCPU. If they do, then system resources are forming a bottleneck and performance will degrade.

M O D P C E X E | Executive Mode Percent

MODPCEXE contains the percentage of time the CPU spent in executive mode during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: MODEXETM - Executive Mode Time

ELEMENT DERIVATION: MODPCEXE is computed using the following algorithm:

```
  if DURATION > 0 then
    MODPCEXE = MODEXETM * 100 / DURATION;
  else
    MODPCEXE = .;
```

RANGE OF VALUE: The value of MODPCEXE ranges from 0 to 100.

SUMMARIZATION PROCESS: MODPCEXE contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: If MODPCEXE consistently exceeds 15 percent to 20 percent of MODPCCPU, the system is probably spending a lot of time serving RMS requests. Consider adjusting file buffers and index caches to reduce time spent in executive mode.

M O D P C I D L | Idle Percent

MODPCIDL contains the percentage time the CPU spent in idle mode during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: MODIDLTM - Idle Time

ELEMENT DERIVATION: MODPCIDL is computed using the following algorithm:

```
  if DURATION > 0 then
    MODPCIDL = MODIDLTM * 100 / DURATION;
  else
    MODPCIDL = .;
```

RANGE OF VALUE: The value of MODPCIDL ranges from 0 to 100.

SUMMARIZATION PROCESS: MODPCIDL contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: MODPCIDL should be close to 100 percent minus MODPCCPU. If a system is a compute server for batch jobs, MODPCIDL may approach 0. For typical time sharing use, however, if MODPCIDL drops below 5 percent consistently, the system is being overtaxed. If MODPCIDL is greater than 10 percent but performance seems inadequate, the system is probably waiting on resources such as page or swap files.

M O D P C I N T | Interrupt Stack Percent

MODPCINT contains the percentage of time spent in interrupt stack mode during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: MODINTTM - Interrupt Stack Time

ELEMENT DERIVATION: MODPCINT is computed using the following algorithm:

```
  if DURATION > 0 then
    MODPCINT = MODINTTM * 100 / DURATION;
  else
```

```
    MODPCINT = .;
```

RANGE OF VALUE: The value of MODPCINT ranges from 0 to 100.

SUMMARIZATION PROCESS: MODPCINT contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: On a nonclustered system, MODPCINT usually indicates a high number of interrupts from peripheral devices such as disks, tapes, printers, and terminals. If MODPCINT exceeds 7 percent and it seems that terminals and printers may be at fault, consider changing the configuration to use terminal servers for those devices.

On a clustered system, MODPCINT may represent time spent serving distributed locks or providing disk and tape services to other members of the cluster. If MODPCINT exceeds 10 percent on a clustered system, monitor the DLOCK, SCS, and MSCP values.

M O D P C K E R | Kernel Mode Percent

MODPCKER contains the percentage of time the CPU spent in kernel mode during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: MODKERTM - Kernel Mode Time

ELEMENT DERIVATION: MODPCKER is computed using the following algorithm:

```
  if DURATION > 0 then
    MODPCKER = MODKERTM * 100 / DURATION;
  else
    MODPCKER = .;
```

RANGE OF VALUE: The value of MODPCKER ranges from 0 to 100.

SUMMARIZATION PROCESS: MODPCKER contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: A high value of MODPCKER (25 percent or more of MODPCCPU) indicates an overuse of some system-wide resources:

- o Memory limitations could be causing paging and swapping.
- o Excessive locking will use global sections and require the interaction of the VMS Lock Manager.
- o Excessive process creation requires allocation of control blocks as well as high amounts of paging.
- o Excessive image activation can cause high page faulting and logical name translation.
- o Excessive file system activity such as file creation, opens and closes, file extending, and file deletions requires kernel mode activities.
- o Excessive DECnet traffic requires kernel mode intervention, even if the packets are not addressed to this node.

M O D P C M P S | Multi-Processor Sync Percent

MODPCMPS contains the percentage of time the CPU spent in multi-processor sync mode during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: MODMPSTM - Multi-Processor Sync Time

ELEMENT DERIVATION: MODPCMPS is computed using the following algorithm:

```
  if DURATION > 0 then
    MODPCMPS = MODMPSTM * 100 / DURATION;
  else
    MODPCMPS = .;
```

RANGE OF VALUE: The value of MODPCMPS ranges from 0 to 100.

SUMMARIZATION PROCESS: MODPCMPS contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: If MODPCMPS is larger than 8 percent of MODPCCPU, the multiprocessor system is spending too much time trying to resolve system resource accesses. Examine the paging, I/O, and locking measurements if this occurs.

```
+-----------------+
| M O D P C S U P |  Supervisor Mode Percent
+-----------------+
MODPCSUP contains the percentage of time spent in supervisor
mode during DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  MODSUPTM - Supervisor Mode Time

ELEMENT DERIVATION:  MODPCSUP is computed using the following
algorithm:

  if DURATION > 0 then
    MODPCSUP = MODSUPTM * 100 / DURATION;
  else
    MODPCSUP = .;

RANGE OF VALUE:  The value of MODPCSUP ranges from 0 to 100.

SUMMARIZATION PROCESS:  MODPCSUP contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  MODPCSUP is the percentage of time spent
executing system services as directly requested from user
processes.  If MODPCSUP is larger than MODPCUSR, the system
is probably spending too much time running DEC Command
Language (DCL) command files.  Because DCL is less efficient
than a compiled program, this should be watched carefully.


+-----------------+
| M O D P C U S R |  User Mode Percent
+-----------------+
MODPCUSR contains the percentage of time spent in user mode
during DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  MODUSRTM - User Mode Time

ELEMENT DERIVATION:  MODPCUSR is computed using the following
algorithm:

  if DURATION > 0 then
    MODPCUSR = MODUSRTM * 100 / DURATION;
  else
    MODPCUSR = .;

RANGE OF VALUE:  The value of MODPCUSR ranges from 0 to 100.

SUMMARIZATION PROCESS:  MODPCUSR contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  MODPCUSR represents the percentage time spent
executing user applications.  MODPCUSR should be more than 30
percent of the value in MODPCCPU in a typically healthy
system.


+-----------------+
| M O D S U P T M |  Supervisor Mode Time
+-----------------+
MODSUPTM contains the number of seconds the CPU spent in
supervisor mode during DURATION (measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is SAS time
value.

SEE ALSO:  MODCPUTM - Total CPU Time
           MODUSRTM - User Mode Time

ELEMENT DERIVATION:  MODSUPTM is set to the value of the
MNR_MOD$L_SUPER field of the VAX MONITOR MODES record minus
the previous value.

RANGE OF VALUE:  MODSUPTM is a SAS time value ranging from
00:00.00 to the size of the DURATION (measurement interval)
value.

SUMMARIZATION PROCESS:  MODSUPTM contains the sum of all
values of MODSUPTM processed in the level of summarization.

USAGE NOTES:  MODSUPTM is time charged to the user accounts;
it represents time spent by the system serving direct
requests for system services.  MODSUPTM is usually smaller
than MODUSRTM, unless the process makes an unusually large
number of requests for system services or users are doing
their processing using large numbers of DCL command files.
Minimize use of DCL for normal processing because it uses too
many resources for interpretation and execution.
```

```
+-----------------+
| M O D U S R T M |  User Mode Time
+-----------------+
MODUSRTM contains the number of seconds the CPU spent in user
mode during DURATION (measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is SAS time
value.

SEE ALSO:  MODCPUTM - Total CPU Time

ELEMENT DERIVATION:  MODUSRTM is set to the value of the
MNR_MOD$L_USER field of the VAX MONITOR MODES record minus
the previous value.

RANGE OF VALUE:  MODUSRTM is a SAS time value ranging from
00:00.00 to the size of the DURATION (measurement interval)
value.

SUMMARIZATION PROCESS:  MODUSRTM contains the sum of all
values of MODUSRTM processed in the level of summarization.

USAGE NOTES:  MODUSRTM represents the amount of time spent
directly executing program instructions in user images.  A
well-tuned and balanced system usually spends 60 percent or
more of its time in the user and supervisor modes and the
majority of that amount in the user mode.


+-----------------+
| M O D V E C T M |  CPU Time of Vector Consumers
+-----------------+
MODVECTM contains the total number of seconds during which
vector consumers were scheduled on the processor.

TYPE:  Accumulated - Numeric, measurement unit is SAS time
value.

SEE ALSO:  MODCPUTM - Total CPU Time

ELEMENT DERIVATION:  MODVECTM is set to the value of the
MNR_MOD$L_TICKS field from the VMS MONITOR VECTOR record,
divided by 100.

RANGE OF VALUE:  MODVECTM is a SAS time value ranging from
00:00.00 to the size of the DURATION (measurement interval)
value.  MODVECTM is reported as zero for processors
identified by VMS as "vector absent".

SUMMARIZATION PROCESS:  MODVECTM contains the sum of all
values of MODVECTM processed in the level of summarization.

USAGE NOTES:  MODVECTM is not a strict measure of the usage
of the vector CPU.  MODVECTM is the time during which a
vector consumer has reserved both the scalar and the vector
CPU for use.
```

B.3.4 - The VMS Monitor Disk (DEMDSK) File

FILE ABSTRACT:	VMS Monitor Disk File
CONTENT	Quantifies the operation of each disk on the VMS system during the monitoring interval
USE	To report on the operation of each disk on the system for performance tuning and capacity planning.
SOURCE	VMS MONITOR record type 12 as documented in the VMS Monitor Reference Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DSKNODE	DSKCTRLR	DSKUNIT	DSKVOL
	YEAR	MONTH	DAY	HOUR	ENDTS
DAYS	SYSID	DSKNODE	DSKCTRLR	DSKUNIT	DSKVOL
	YEAR	MONTH	DAY	HOUR	
WEEKS	SYSID	DSKNODE	DSKCTRLR	DSKUNIT	DSKVOL
	YEAR	WEEK	ZONE	HOUR	
MONTHS	SYSID	DSKNODE	DSKCTRLR	DSKUNIT	DSKVOL
	YEAR	MONTH	ZONE		
YEARS	N/A				

TABLES	N/A

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XDWM..E	DSKCTRLR	- Disk Controller
XDWM..E	DSKNODE	- Nodename
XDWM..E	DSKUNIT	- Disk Unit Number
XDWM..E	DSKVOL	- Disk Volume Name
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
XDWM..E	ZONE	- Time Zone
Common Data Elements		

```
 XD....E  DAYNAME  - Name of Day of Week
 XDWM..   DEXBOOTS - Monitor System Boot Time Stamp
 XDWM..E  DEXDUR   - Requested Recording Interval Time
 X.....E  DEXINTNO - Reserved Field Incremented by Monitor
 X.....E  DEXMONVR - VMS Monitor Version
 XDWM..E  DEXTS    - Monitor Time Stamp
 XDWM..E  DURATION - Recording Interval Time
 XDWM..E  ENDTS    - End Time Stamp
 XDWM..E  INTERVLS - Number of Recording Intervals
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  STARTTS  - Start Time Stamp

Retained Data Elements

 X.....   DSKALLOC - Allocation Class
 X.....   DSKFLAGS - Flags

Accumulated Data Elements

 XDWM..E  DSKIOOPS - Disk I/O Operations
 XDWM..E  DSKIOQUE - Disk I/O Queue Length Samples Sum

Derived Data Elements

 XDWM..E  DSKAVRES - Disk Average Response Time
 XDWM..E  DSKPSIO  - Disk I/O Operation Rate
```

D S K A L L O C — Allocation Class

DSKALLOC contains the allocation class code for the disk.

TYPE: Retained - Numeric, measurement unit is integer, 1 byte long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DSKALLOC is set to the value of the MNR_DSK$B_ALLOCLS field of the VMS MONITOR DISK record.

RANGE OF VALUE: DSKALLOC takes any integer value from 0 to 255.

SUMMARIZATION PROCESS: DSKALLOC contains the last value of DSKALLOC processed in the level of summarization.

USAGE NOTES: None

D S K A V R E S — Disk Average Response Time

DSKAVRES contains the average response time for the disk.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: DSKIOOPS - Disk I/O Operations
DSKIOQUE - Disk I/O Queue Length Samples Sum
DSKPSIO - Disk I/O Operation Rate

ELEMENT DERIVATION: DSKAVRES is computed using the following algorithm:

```
  if DSKIOOPS > 0 then
    DSKAVRES = DSKIOQUE / DSKIOOPS
  else
    DSKAVRES = .
```

Disk average response time is derived using the following formula:

```
   average response time =
       average queue length / average I/O operation rate
```

where:

```
   average queue length = DSKIOQUE / (DURATION * samplerate)

   average I/O operation rate = DSKIOOPS / DURATION
```

Since the queue length sample rate (samplerate) for VMS MONITOR is 1 per second, the formulas reduce to:

```
   average queue length = DSKIOQUE / DURATION

   average response time = DSKIOQUE / DSKIOOPS
```

RANGE OF VALUE: DSKAVRES takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: DSKAVRES contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: None

| D S K C T R L R | Disk Controller

DSKCTRLR contains the character string that describes the controller of the disk.

TYPE: Retained - Character, length 3.

SEE ALSO: DSKUNIT - Disk Unit Number

ELEMENT DERIVATION: DSKCTRLR is set to the value of the MNR_DSK$T_CTRLR field of the VMS MONITOR DISK record.

RANGE OF VALUE: Not applicable

SUMMARIZATION PROCESS: DSKCTRLR is not summarized. DSKCTRLR is used to determine a unique key for records within a file based on the time-span so other elements in the file can be summarized.

USAGE NOTES: The first letter of the controller is usually D to indicate a disk device. The second letter is the type of bus to which the controller is attached, U for Unibus, B for BI, R for DSSI, etc. The third letter is the sequence letter for the controller on the bus. Controllers start with A and go on.

| D S K F L A G S | Flags

DSKFLAGS contains the flags register for the disk.

TYPE: Retained - Hexadecimal number, 1 byte long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DSKFLAGS is set to the value of the MNR_DSK$B_FLAGS field of the VMS MONITOR DISK record.

RANGE OF VALUE: DSKFLAGS takes any hexadecimal integer value from 00 to FF.

SUMMARIZATION PROCESS: DSKFLAGS contains the last value of DSKFLAGS processed in the level of summarization.

USAGE NOTES: If the low bit is set, this disk is served by the MSCP server.

| D S K I O O P S | Disk I/O Operations

DSKIOOPS contains the number of I/O operations to this disk during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: DSKPSIO - Disk I/O Operation Rate

ELEMENT DERIVATION: DSKIOOPS is set to the value of the MNR_DSK$L_OPCNT field of the VMS MONITOR DISK record.

RANGE OF VALUE: DSKIOOPS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: DSKIOOPS contains the sum of all values of DSKIOOPS processed in the level of summarization.

USAGE NOTES: DSKIOOPS is the number of operations received by the controller for this disk. If some disks are receiving significantly more I/O requests than others, consider moving some of the high-activity files to the more inactive disks.

| D S K I O Q U E | Disk I/O Queue Length Samples Sum

DSKIOQUE contains the sum of the I/O queue length samples for the measurement interval (DURATION).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: DSKAVRES - Disk Average Response Time

ELEMENT DERIVATION: DSKIOQUE is set to the value of the MNR_DSK$L_IOQUELN field of the VMS MONITOR DISK record.

RANGE OF VALUE: DSKIOQUE takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: DSKIOQUE contains the sum of all values of DSKIOQUE processed in the level of summarization.

USAGE NOTES: If DSKIOQUE is almost always 0 for a particular disk, that disk is underutilized as far as I/O activity is concerned. If DSKIOQUE is consistently greater than 2, the disk is probably overly busy. Active files should be moved from more active disks to less active disks to even out the

I/O queues.

To investigate average queue length, use the following formula:

```
average queue length = DSKIOQUE / (DURATION * samplerate)
```

This divides the sum of the queue length sample measurements by the number of samples taken during the period of DURATION. The published queue length sample rate (samplerate) for VMS MONITOR is 1 per second, thus reducing the formula to:

```
average queue length = DSKIOQUE / DURATION
```

The value for samplerate is not provided in VMS MONITOR data.

D S K N O D E | Nodename

DSKNODE contains the string representing the cluster node name for where this disk is located.

TYPE: Retained - Character, length 7.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DSKNODE is set to the value of the MNR_DSK$T_NODENAME field of the VMS MONITOR DISK record.

RANGE OF VALUE: DSKNODE can contain any combination of 1 to 7 alphanumeric characters, including the symbols '_' and '$'.

SUMMARIZATION PROCESS: DSKNODE is not summarized. DSKNODE is used to determine a unique key for records within a file based on the time-span so other elements in the file can be summarized.

USAGE NOTES: DSKNODE helps to identify the disk when cluster-wide statistics are being taken.

D S K P S I O | Disk I/O Operation Rate

DSKPSIO contains the I/O operation rate for the disk.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: DSKIOOPS - Disk I/O Operations

ELEMENT DERIVATION: DSKPSIO is computed using the following algorithm:

```
if DURATION > 0 then
  DSKPSIO = DSKIOOPS / DURATION
else
  DSKPSIO = 0
```

RANGE OF VALUE: DSKPSIO takes any integer value from 0 to DSKIOOPS.

SUMMARIZATION PROCESS: DSKPSIO contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: The values of DSKPSIO supported with acceptable performance varies with the disk drive, controller, and bus being used. What is more important is relative load. If two drives are of the same type and share a controller or are on the same type of controller, DSKPSIO should be the same on average. If one is much larger than the other, the load should probably be balanced by placing some of the more active files from the busier disk onto the less used disk.

D S K U N I T | Disk Unit Number

DSKUNIT contains the unit number of the disk on the controller.

TYPE: Retained - Numeric, measurement unit is integer, 2 bytes long.

SEE ALSO: DSKCTRLR - Disk Controller

ELEMENT DERIVATION: DSKUNIT is set to the value of the MNR_DSK$W_UNITNO field of the VMS MONITOR DISK record.

RANGE OF VALUE: DSKUNIT takes any integer value from 0 to 32,767.

SUMMARIZATION PROCESS: DSKUNIT is not summarized. DSKUNIT is used to determine a unique key for records within a file based on the time-span so other elements in the file can be summarized.

USAGE NOTES: DSKCTRLR and DSKUNIT uniquely identify the disk on a node. If the system is clustered, DSKNODE added will give a unique cluster-wide identification.

D S K V O L	Disk Volume Name

DSKVOL contains the string representing the volume name for the disk.

TYPE: Retained - Character, length 12.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DSKVOL is set to the value of the MNR_DSK$L_NAME field of the VMS MONITOR DISK record.

RANGE OF VALUE: DSKVOL can contain any combination of 1 to 12 alphanumeric characters, including the symbols '_' and '$'.

SUMMARIZATION PROCESS: DSKVOL is not summarized. DSKVOL is used to determine a unique key for records within a file based on the time-span so other elements in the file can be summarized.

USAGE NOTES: The volume name is chosen when the disk was initialized as an identifier of the volume. If the disk is not mounted, the volume name must be known to mount the disk without using elevated privileges (PHYSIO). DSKVOL is not necessarily unique on a system.

B.3.5 - The VMS Monitor SCS (DEMSCS) File

FILE ABSTRACT:	VMS Monitor SCS File
CONTENT	Quantifies the operation of the VMS systems communications services during the monitoring interval.
USE	To report on the operation systems communications services for performance tuning and capacity planning.
SOURCE	VMS MONITOR record type 15 as documented in the VMS Monitor Reference Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	SCSNODE	YEAR	MONTH	DAY
	HOUR	ENDTS			
DAYS	SYSID	SCSNODE	YEAR	MONTH	DAY
	HOUR				
WEEKS	SYSID	SCSNODE	YEAR	WEEK	ZONE
	HOUR				
MONTHS	SYSID	SCSNODE	YEAR	MONTH	ZONE
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year
XDWM..E	SCSNODE	- Node Identifier
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
XDWM..E	ZONE	- Time Zone
Common Data Elements		
XD....E	DAYNAME	- Name of Day of Week
XDWM..	DEXBOOTS	- Monitor System Boot Time Stamp
XDWM..E	DEXDUR	- Requested Recording Interval Time
X.....E	DEXINTNO	- Reserved Field Incremented by Monitor

```
 X.....E  DEXMONVR - VMS Monitor Version
 XDWM..E  DEXTS    - Monitor Time Stamp
 XDWM..E  DURATION - Recording Interval Time
 XDWM..E  ENDTS    - End Time Stamp
 XDWM..E  INTERVLS - Number of Recording Intervals
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  STARTTS  - Start Time Stamp
```

Accumulated Data Elements

```
 XDWM..E  SCSBTKBM - Block Transfer Kilobytes Mapped
 XDWM..   SCSBTRD  - Block Transfer Request Datas
 XDWM..   SCSBTSND - Block Transfer Send Datas
 XDWM..   SCSCQBFR - Connections Queued for Buffer Descript
 XDWM..   SCSCQSND - Connections Queued for Send Credit
 XDWM..E  SCSDGDIS - Datagrams Discarded
 XDWM..E  SCSDGRCV - Datagrams Received
 XDWM..E  SCSDGSNT - Datagrams Sent
 XDWM..   SCSKBRCV - Kilobytes Received by Request Datas
 XDWM..   SCSKBSND - Kilobytes Sent by Send Datas
 XDWM..E  SCSMSGRC - Sequenced Messages Received
 XDWM..E  SCSMSGSN - Sequenced Messages Sent
```

Derived Data Elements

```
 XDWM..E  SCSPSKBT - Block Transfer Kilobytes Mapped Rate
```

S C S B T K B M — Block Transfer Kilobytes Mapped

SCSBTKBM contains the number of kilobytes mapped into buffers for block transfer between nodes.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SCSKBSND - Kilobytes Sent by Send Datas
SCSKBRCV - Kilobytes Received by Request Datas

ELEMENT DERIVATION: SCSBTKBM is set to the value of the MNR_SCS$L_KBYTMAPD field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSBTKBM takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSBTKBM contains the sum of all values of SCSBTKBM processed in the level of summarization.

USAGE NOTES: SCSBTKBM is the best means of estimating the data transferred between the recording node and remote nodes. Before any data can be transferred, buffers have to be mapped for the transferring process. SCSBTKBM is the number of kilobytes mapped into these buffers.

S C S B T R D — Block Transfer Request Datas

SCSBTRD contains the number of data-block transfer requests that originated by the current recording node and targeted to remote nodes.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SCSKBRCV - Kilobytes Received by Request Datas

ELEMENT DERIVATION: SCSBTRD is set to the value of the MNR_SCS$L_REQDATS field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSBTRD takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSBTRD contains the sum of all values of SCSBTRD processed in the level of summarization.

USAGE NOTES: There is no direct correlation from this value to the number of kilobytes transferred. Assuming that all SCSBTRDs are honored, the average transfer block could be determined by dividing SCSBTRD into SCSKBRCV.

S C S B T S N D — Block Transfer Send Datas

SCSBTSND contains the number of data block sends initiated on the recording node targeted to remote nodes.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SCSKBSND - Kilobytes Sent by Send Datas

ELEMENT DERIVATION: SCSBTSND is set to the value of the MNR_SCS$L_SNDATS field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSBTSND takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSBTSND contains the sum of all values of SCSBTSND processed in the level of summarization.

USAGE NOTES: SCSBTSND is the number of times the recording node notifies another node that it has some data for it. To find the average block size, divide SCSBTSND into SCSKBSND.

S C S C Q B F R Connections Queued for Buffer Descript

SCSCQBFR contains the number of connections waiting for buffers on the recording node.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SCSCQSND - Connections Queued for Send Credit

ELEMENT DERIVATION: SCSCQBFR is set to the value of the MNR_SCS$L_QBDTCNT field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSCQBFR takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSCQBFR contains the sum of all values of SCSCQBFR processed in the level of summarization.

USAGE NOTES: SCSCQBFR are connections waiting for buffers on the recording node to initiate data transfers. If this value is much greater than 0, cluster data transfer could be held up because of an inadequate number of buffers. To increase the number of buffers, increase the SYSGEN parameter SCSBUFFCNT for the recording node.

S C S C Q S N D Connections Queued for Send Credit

SCSCQSND contains the number of connections waiting for buffers on the remote node.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SCSCQBFR - Connections Queued for Buffer Descript

ELEMENT DERIVATION: SCSCQSND is set to the value of the MNR_SCS$L_QCRCNT field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSCQSND takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSCQSND contains the sum of all values of SCSCQSND processed in the level of summarization.

USAGE NOTES: SCSCQSND are connections waiting for buffers on the remote node to initiate data transfers. If this value is much greater than 0 cluster data transfer could be held up because of an inadequate number of buffers. To increase the number of buffers, increase the SYSGEN parameter SCSBUFFCNT on the remote nodes.

S C S D G D I S Datagrams Discarded

SCSDGDIS contains the count of datagrams that the CI port discarded.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SCSDGDIS is set to the value of the MNR_SCS$L_DGDISCARD field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSDGDIS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSDGDIS contains the sum of all values of SCSDGDIS processed in the level of summarization.

USAGE NOTES: If the CI port cannot process a datagram it is discarded and the process is notified. SCSDGDIS is the count of those occurrences.

S C S D G R C V Datagrams Received

SCSDGRCV contains the number of data packets received by SCS.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SCSDGSNT - Datagrams Sent

ELEMENT DERIVATION: SCSDGRCV is set to the value of the MNR_SCS$L_DGRCVD field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSDGRCV takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSDGRCV contains the sum of all values of SCSDGRCV processed in the level of summarization.

USAGE NOTES: SCSDGRCV is a count of all data packets received regardless of the source of the request. The average transfer size of received blocks is not readily available. The average transfer size for all transfers is:
  SCSBTKBM / (SCSDGRCV + SCSDGSNT)

S C S D G S N T | Datagrams Sent

SCSDGSNT contains the number of data packets sent by SCS.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SCSDGRCV - Datagrams Received

ELEMENT DERIVATION: SCSDGSNT is set to the value of the MNR_SCS$L_DGSENT field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSDGSNT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSDGSNT contains the sum of all values of SCSDGSNT processed in the level of summarization.

USAGE NOTES: SCSDGSNT is a count of all data packets sent regardless of the source of the request. The average transfer size of sent blocks is not readily available. The average transfer size for all transfers is:
  SCSBTKBM / (SCSDGRCV + SCSDGSNT)

S C S K B R C V | Kilobytes Received by Request Datas

SCSKBRCV contains the number of kilobytes received in answer to data-block transfer requests sent by the recording node.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SCSBTRD - Block Transfer Request Datas

ELEMENT DERIVATION: SCSKBRCV is set to the value of the MNR_SCS$L_KBYTREQD field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSKBRCV takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSKBRCV contains the sum of all values of SCSKBRCV processed in the level of summarization.

USAGE NOTES: SCSBTRD is the number of requests for data transfers by the recording node. There is no direct correlation from this value to the number of kilobytes transferred. Assuming that all SCSBTRDs are honored, the average transfer block could be determined by dividing SCSBTRD into SCSKBRCV.

S C S K B S N D | Kilobytes Sent by Send Datas

SCSKBSND contains the number of kilobytes sent by data block sends initiated on the recording node targeted to remote nodes.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SCSBTSND - Block Transfer Send Datas

ELEMENT DERIVATION: SCSKBSND is set to the value of the MNR_SCS$L_KBYTSENT field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSKBSND takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSKBSND contains the sum of all values of SCSKBSND processed in the level of summarization.

USAGE NOTES: SCSBTSND is the number of times the recording node notifies another node that it has some data for it. To find the average block size, divide SCSBTSND into SCSKBSND.

S C S M S G R C | Sequenced Messages Received

SCSMSGRC contains the number of sequencing messages the recording node receives from other nodes.

TYPE: Accumulated - Numeric, measurement unit is integer, 4

bytes long.

SEE ALSO: SCSMSGSN - Sequenced Messages Sent

ELEMENT DERIVATION: SCSMSGRC is set to the value of the MNR_SCS$L_MSGRCVD field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSMSGRC takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSMSGRC contains the sum of all values of SCSMSGRC processed in the level of summarization.

USAGE NOTES: SCSMSGRC is a count of the messages sent by other nodes for communicating with MSCP devices and the lock manager on the recording node. This is part of the overhead of having cluster-wide resources served by the recording node.

S C S M S G S N | Sequenced Messages Sent

SCSMSGSN contains the number of sequencing messages sent to other nodes.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SCSMSGRC - Sequenced Messages Received

ELEMENT DERIVATION: SCSMSGSN is set to the value of the MNR_SCS$L_MSGSENT field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSMSGSN takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSMSGSN contains the sum of all values of SCSMSGSN processed in the level of summarization.

USAGE NOTES: SCSMSGSN is a count of the messages sent by the recording node for communicating with MSCP devices and the lock managers on other node. This is part of the overhead of having cluster-wide resources served by other nodes.

S C S N O D E | Node Identifier

SCSNODE contains the name of the node for the recording interval.

TYPE: Retained - Character, length 7. Display format is character.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SCSNODE is set to the value of the MNR_SCS$T_NODENAME field of the VMS MONITOR SCS record.

RANGE OF VALUE: SCSNODE can contain any combination of 1 to 7 alphanumeric characters, including the symbols '_' and '$'.

SUMMARIZATION PROCESS: SCSNODE is not summarized. SCSNODE is used to determine a unique key for records within a file based on the time-span so other elements in the file can be summarized.

USAGE NOTES: None

S C S P S K B T | Block Transfer Kilobytes Mapped Rate

SCSPSKBT contains the number of kilobytes mapped per second for block transfers.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SCSBTKBM - Block Transfer Kilobytes Mapped

ELEMENT DERIVATION: SCSPSKBT is computed using the following algorithm:

```
if DURATION > 0 then
  SCSPSKBT = SCSBTKBM / DURATION
else
  SCSPSKBT = 0
```

RANGE OF VALUE: SCSPSKBT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SCSPSKBT contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SCSPSKBT is the best means of estimating the data transfer rate between the recording node and remote nodes. Before any data can be transferred, buffers have to be mapped for the transferring process. SCSBTKBM is the number of kilobytes mapped into buffers, and SCSPSKBT is the rate of mapping.

B.3.6 - The VMS Monitor RMS (DEMRMS) File

FILE ABSTRACT:	VMS Monitor RMS File
CONTENT	Quantifies the VMS record management services during the monitoring interval
USE	To report on usage including get bytes, put bytes, update bytes, deletes, finds, connects and disconnects, for performance tuning and capacity planning.
SOURCE	VMS MONITOR record type 20 as documented in the VMS Monitor Reference Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	RMSFILE	YEAR	MONTH	DAY
	HOUR	ENDTS			
DAYS	SYSID	RMSFILE	YEAR	MONTH	DAY
	HOUR				
WEEKS	SYSID	RMSFILE	YEAR	WEEK	ZONE
	HOUR				
MONTHS	SYSID	RMSFILE	YEAR	MONTH	ZONE
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year
XDWM..E	RMSFILE	- File Number
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
XDWM..E	ZONE	- Time Zone
Common Data Elements		
XD....E	DAYNAME	- Name of Day of Week
XDWM..	DEXBOOTS	- Monitor System Boot Time Stamp

```
 XDWM..E  DEXDUR   - Requested Recording Interval Time
 X.....E  DEXINTNO - Reserved Field Incremented by Monitor
 X.....E  DEXMONVR - VMS Monitor Version
 XDWM..E  DEXTS    - Monitor Time Stamp
 XDWM..E  DURATION - Recording Interval Time
 XDWM..E  ENDTS    - End Time Stamp
 XDWM..E  INTERVLS - Number of Recording Intervals
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  STARTTS  - Start Time Stamp

Retained Data Elements

 X.....E  RMSFILNM - File Name
 X.....E  RMSORG   - Organization

Accumulated Data Elements

 XDWM..   RMSALAST - File Append Lock Blocking ASTs
 XDWM..   RMSALCNV - File Append Lock Conversions
 XDWM..   RMSALDEQ - File Append Lock Dequeues
 XDWM..   RMSALENQ - File Append Lock Enqueues
 XDWM..   RMSBUCKT - File Bucket Splits
 XDWM..E  RMSCLOSE - File Closes
 XDWM..E  RMSCONN  - File Connects
 XDWM..E  RMSDELET - File Deletes
 XDWM..E  RMSDISCN - File Disconnects
 XDWM..E  RMSEXTBL - File Extend Blocks
 XDWM..E  RMSEXTEN - File Extends
 XDWM..   RMSFBAST - File Lock Blocking ASTs
 XDWM..E  RMSFLUSH - File Flushes
 XDWM..   RMSGBAST - File Global Buffer Blocking ASTs
 XDWM..   RMSGBCNV - File Global Buffer Conversions
 XDWM..   RMSGBDEQ - File Global Buffer Dequeues
 XDWM..   RMSGBENQ - File Global Buffer Enqueues
 XDWM..   RMSGBRIO - File Global Buffer Read I/Os
 XDWM..   RMSGBWIO - File Global Buffer Write I/Os
 XDWM..   RMSGCATT - File Global Cache Attempts
 XDWM..   RMSGCHIT - File Global Cache Hits
 XDWM..E  RMSGETBT - File GET Bytes
 XDWM..   RMSGSCNV - File Global Section Conversions
 XDWM..   RMSGSDEQ - File Global Section Dequeues
 XDWM..   RMSGSENQ - File Global Section Enqueues
 XDWM..E  RMSKEYFD - Keyed Finds
 XDWM..E  RMSKEYGT - Keyed GETs
 XDWM..E  RMSKEYPT - Keyed PUTs
 XDWM..   RMSLBAST - File Local Buffer Blocking ASTs
 XDWM..   RMSLBCNV - File Local Buffer Conversions
 XDWM..   RMSLBDEQ - File Local Buffer Dequeues
 XDWM..   RMSLBENQ - File Local Buffer Enqueues
 XDWM..   RMSLBRIO - File Local Buffer Read I/Os
 XDWM..   RMSLBWIO - File Local Buffer Write I/Os
 XDWM..   RMSLCATT - File Local Cache Attempts
 XDWM..   RMSLCHIT - File Local Cache Hits
 XDWM..   RMSLKCNV - File Lock Conversions
 XDWM..   RMSLKDEQ - File Lock Dequeues
 XDWM..   RMSLKENQ - File Lock Enqueues
 XDWM..   RMSMULTB - File Multibucket Splits
 XDWM..E  RMSOPEN  - File Opens
 XDWM..E  RMSPUTBT - File PUT Bytes
 XDWM..E  RMSRDBT  - File Read Bytes
 XDWM..E  RMSREAD  - File Reads
 XDWM..E  RMSREWND - File Rewinds
 XDWM..E  RMSRFAFD - Record File Address Finds
 XDWM..E  RMSRFAGT - Record File Address GETs
 XDWM..   RMSRLCNV - Record Lock Conversions
 XDWM..   RMSRLDEQ - Record Lock Dequeues
 XDWM..   RMSRLENQ - Record Lock Enqueues
 XDWM..E  RMSSEQFD - Sequential Finds
 XDWM..E  RMSSEQGT - Sequential GETs
 XDWM..E  RMSSEQPT - Sequential PUTs
 XDWM..E  RMSTRUBL - File Truncate Blocks
 XDWM..E  RMSTRUNC - File Truncates
 XDWM..E  RMSUPDBT - File Update Bytes
 XDWM..E  RMSUPDTE - File Updates
 XDWM..E  RMSWRBT  - File Write Bytes
 XDWM..E  RMSWRITE - File Writes


+-----------------+
| R M S A L A S T |  File Append Lock Blocking ASTs
+-----------------+
RMSALAST contains the number of append lock blocking
asynchronous system traps (ASTs) for the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSFBAST - File Lock Blocking ASTs
          RMSGBAST - File Global Buffer Blocking ASTs
          RMSLBAST - File Local Buffer Blocking ASTs
          RMSALENQ - File Append Lock Enqueues
          RMSALDEQ - File Append Lock Dequeues
          RMSALCNV - File Append Lock Conversions

ELEMENT DERIVATION: RMSALAST is set to the value of the
MNR_RMS$L_APPBLKAST field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSALAST takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: RMSALAST contains the sum of all
values of RMSALAST processed in the level of summarization.

USAGE NOTES: An AST is a mechanism for signaling when an
```

asynchronous event has been completed. RMSALAST is a count of such traps used to signal processes that they can now attempt to append to the file.

R M S A L C N V | File Append Lock Conversions

RMSALCNV contains the number of append lock conversions for the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSALENQ - File Append Lock Enqueues
RMSALDEQ - File Append Lock Dequeues

ELEMENT DERIVATION: RMSALCNV is set to the value of the MNR_RMS$L_APPLCKCNVS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSALCNV takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSALCNV contains the sum of all values of RMSALCNV processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed its processing, it then sends a dequeue to free the resource. RMSALCNV is the number of append locks granted on this file for DURATION (measurement interval).

R M S A L D E Q | File Append Lock Dequeues

RMSALDEQ contains the number of append lock dequeues requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSALENQ - File Append Lock Enqueues
RMSALCNV - File Append Lock Conversions

ELEMENT DERIVATION: RMSALDEQ is set to the value of the MNR_RMS$L_LCKDEQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSALDEQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSALDEQ contains the sum of all values of RMSALDEQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed its processing, it then sends a dequeue to free the resource. RMSALDEQ is the number of append locks freed on this file for DURATION (measurement interval).

R M S A L E N Q | File Append Lock Enqueues

RMSALENQ contains the number of append lock enqueues requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSALDEQ - File Append Lock Dequeues
RMSALCNV - File Append Lock Conversions

ELEMENT DERIVATION: RMSALENQ is set to the value of the MNR_RMS$L_APPLCKENQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSALENQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSALENQ contains the sum of all values of RMSALENQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed its processing, it then sends a dequeue to free the resource. RMSALENQ is the number of append locks requested on this file for DURATION (measurement interval).

R M S B U C K T File Bucket Splits

RMSBUCKT contains the number of bucket splits performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSMULTB - File Multibucket Splits

ELEMENT DERIVATION: RMSBUCKT is set to the value of the MNR_RMS$L_BKTSPLT field of the VMS MONITOR RMS record

RANGE OF VALUE: RMSBUCKT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSBUCKT contains the sum of all values of RMSBUCKT processed in the level of summarization.

USAGE NOTES: Bucket splits occur during insert operations, when any of the index buffers (buckets) fill up to the level defined in the file header parameter PCTFULL. This happens when keys are inserted between keys in a full index buffer. Split buckets increase file processing time since the process must search for the subsequent indexes in another buffer. It is also likely that when a key search is occurring, the needed bucket will not be in the buffers, causing physical I/Os to continue the search. When buffer space is limited, split buckets often lead to index buffer thrashing, where buckets are read in, overwriting other buckets that will be needed immediately. When a file is severely split, use the VMS CONVERT utility to refill the buckets evenly.

R M S C L O S E File Closes

RMSCLOSE contains the number of times the file was closed in DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSCLOSE is set to the value of the MNR_RMS$L_CLOSES field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSCLOSE takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSCLOSE contains the sum of all values of RMSCLOSE processed in the level of summarization.

USAGE NOTES: File closes cost almost nothing unless the file size changed significantly or they are part of an open-close-open cycle that gets out of hand.

R M S C O N N File Connects

RMSCONN contains the count of $CONNECTs on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSOPEN - File Opens
RMSDISCN - File Disconnects

ELEMENT DERIVATION: RMSCONN is set to the value of the MNR_RMS$L_CONNECTS field of the VMS MONITOR RMS record

RANGE OF VALUE: RMSCONN takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSCONN contains the sum of all values of RMSCONN processed in the level of summarization.

USAGE NOTES: After a process opens a file, to process I/O operations on it the process must connect the access blocks with the file control blocks. This allows the process to use the features of RMS such as file and record locking and keyed searches.

R M S D E L E T File Deletes

RMSDELET contains the number of $DELETEs performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSDELET is set to the value of the MNR_RMS$L_DELETES field of the VMS MONITOR RMS record

RANGE OF VALUE: RMSDELET takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSDELET contains the sum of all values of RMSDELET processed in the level of summarization.

USAGE NOTES: RMSDELET is incremented every time a record is deleted from an RMS file. If the file is indexed, the index buckets must be updated as well as the data.

R M S D I S C N | File Disconnects

RMSDISCN contains the number of $DISCONNECTs performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSCONN - File Connects

ELEMENT DERIVATION: RMSDISCN is set to the value of the MNR_RMS$L_DISCONNECTS field of the VMS MONITOR RMS record

RANGE OF VALUE: RMSDISCN takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSDISCN contains the sum of all values of RMSDISCN processed in the level of summarization.

USAGE NOTES: When several means of accessing a file are going to be used, connecting and disconnecting the record access blocks that access it in the different ways is the most efficient means.

R M S E X T B L | File Extend Blocks

RMSEXTBL contains the number of extend blocks in the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSEXTEN - File Extends

ELEMENT DERIVATION: RMSEXTBL is set to the value of the MNR_RMS$L_EXTBLOCKS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSEXTBL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSEXTBL contains the sum of all values of RMSEXTBL processed in the level of summarization.

USAGE NOTES: When a file is first created, there is a certain number of blocks of data space allocated for it. As the file grows beyond the original allocation, new areas, called extents are allocated for the new data. RMSEXTBL is the total number of blocks by which the file has been extended.

R M S E X T E N | File Extends

RMSEXTEN contains the number of extents given to the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSEXTBL - File Extend Blocks

ELEMENT DERIVATION: RMSEXTEN is set to the value of the MNR_RMS$L_EXTENDS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSEXTEN takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSEXTEN contains the sum of all values of RMSEXTEN processed in the level of summarization.

USAGE NOTES: When a file is first created, there is a certain number of blocks of data space allocated for it. As the file grows beyond the original allocation, new areas, called extents are allocated for the new data. RMSEXTEN is the total number of group of blocks by which the file has been extended during the recording interval.

R M S F B A S T | File Lock Blocking ASTs

RMSFBAST contains the number of file lock blocking asynchronous system traps (ASTs) for the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSGBAST - File Global Buffer Blocking ASTs
RMSLBAST - File Local Buffer Blocking ASTs
RMSALAST - File Append Lock Blocking ASTs

ELEMENT DERIVATION: RMSFBAST is set to the value of the MNR_RMS$L_GBLBLKAST field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSFBAST takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSFBAST contains the sum of all values of RMSFBAST processed in the level of summarization.

USAGE NOTES: An AST is a mechanism for signaling when an asynchronous event has been completed. RMSFBAST is a count of such traps used to signal processes that they can now attempt to lock the file.

R M S F I L E | File Number

RMSFILE contains the sequential number of the file.

TYPE: Retained - Numeric, measurement unit is integer, 1 byte long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSFILE is set to the value of the MNR_RMS$B_FILNUM field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSFILE takes any integer value from 1 to 255.

SUMMARIZATION PROCESS: RMSFILE is not summarized. RMSFILE is used to determine a unique key for records within a file based on the time-span so other elements in the file can be summarized.

USAGE NOTES: RMSFILE is an internal sequencing number for VMS MONITOR to keep track of the files on which it is recording data.

R M S F I L N M | File Name

RMSFILNM contains the file specification associated with the RMSFILE Number recorded in the MONITOR RMS record.

TYPE: Retained - Character, length 80.

SEE ALSO: RMSFILE - File Number

ELEMENT DERIVATION: RMSFILNM is set to the contents of the RMS File Name field, MNR_FIL$T_FILENAME, from the VMS MONITOR RMS File record (type 131).

RANGE OF VALUE: RMSFILNM can contain any combination of 1 to 80 alphanumeric characters.

SUMMARIZATION PROCESS: RMSFILNM contains the last value of RMSFILNM processed in the level of summarization.

USAGE NOTES: RMS File Name is recorded in the CA MICS VMX MONITOR RMS (DEMRMS) file for up to two file names provided from the MONITOR type 131 record.

R M S F L U S H | File Flushes

RMSFLUSH contains the number of $FLUSHes performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSFLUSH is set to the value of the MNR_RMS$L_FLUSHES field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSFLUSH takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSFLUSH contains the sum of all values of RMSFLUSH processed in the level of summarization.

USAGE NOTES: RMSFLUSH is the count of times that the buffers were written to disk by specific request of the user.

```
+-----------------+
| R M S G B A S T |  File Global Buffer Blocking ASTs
+-----------------+
RMSGBAST contains the number of global buffer blocking
asynchronous system traps (ASTs) for the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSFBAST - File Lock Blocking ASTs
          RMSLBAST - File Local Buffer Blocking ASTs
          RMSALAST - File Append Lock Blocking ASTs

ELEMENT DERIVATION: RMSGBAST is set to the value of the
MNR_RMS$L_GBLBLKAST field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGBAST takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: RMSGBAST contains the sum of all
values of RMSGBAST processed in the level of summarization.

USAGE NOTES: An AST is a mechanism for signaling when an
asynchronous event has been completed.  RMSGBAST is a count
of such traps used to signal processes that they can now
attempt to access the global buffers of the file.


+-----------------+
| R M S G B C N V |  File Global Buffer Conversions
+-----------------+
RMSGBCNV contains the number of global buffer conversions for
the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSGBENQ - File Global Buffer Enqueues
          RMSGBDEQ - File Global Buffer Dequeues

ELEMENT DERIVATION: RMSGBCNV is set to the value of the
MNR_RMS$L_GBLCKCNVS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGBCNV takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: RMSGBCNV contains the sum of all
values of RMSGBCNV processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a
resource in VMS, it issues an enqueue on that resource.  When
the resource becomes available, VMS converts the enqueue into
an exclusive lock for the requesting process.  When the
process has completed its processing, it then sends a dequeue
to free the resource.  RMSGBCNV is the number of locks
granted on global buffers for this file for DURATION
(measurement interval).


+-----------------+
| R M S G B D E Q |  File Global Buffer Dequeues
+-----------------+
RMSGBDEQ contains the number of global buffer dequeues
requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSGBENQ - File Global Buffer Enqueues
          RMSGBCNV - File Global Buffer Conversions

ELEMENT DERIVATION: RMSGBDEQ is set to the value of the
MNR_RMS$L_GBLCKDEQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGBDEQ takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: RMSGBDEQ contains the sum of all
values of RMSGBDEQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a
resource in VMS, it issues an enqueue on that resource.  When
the resource becomes available, VMS converts the enqueue into
an exclusive lock for the requesting process.  When the
process has completed its processing, it then sends a dequeue
to free the resource.  RMSGBDEQ is the number of global
buffer locks freed on this file for DURATION (measurement
interval).


+-----------------+
| R M S G B E N Q |  File Global Buffer Enqueues
+-----------------+
RMSGBENQ contains the number of global buffer enqueues
requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSGBDEQ - File Global Buffer Dequeues
```

RMSGBCNV - File Global Buffer Conversions

ELEMENT DERIVATION: RMSGBENQ is set to the value of the MNR_RMS$L_GBLCKENQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGBENQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSGBENQ contains the sum of all values of RMSGBENQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed its processing, it then sends a dequeue to free the resource. RMSGBENQ is the number of global buffer locks requested on this file for DURATION (measurement interval).

R M S G B R I O — File Global Buffer Read I/Os

RMSGBRIO contains the number of global buffer read I/Os performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSGBWIO - File Global Buffer Write I/Os
RMSLBRIO - File Local Buffer Read I/Os
RMSLBWIO - File Local Buffer Write I/Os

ELEMENT DERIVATION: RMSGBRIO is set to the value of the MNR_RMS$L_GBRDIRIOS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGBRIO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSGBRIO contains the sum of all values of RMSGBRIO processed in the level of summarization.

USAGE NOTES: The data read from or written to an RMS file is kept in buffers until the user process requests it or RMS is ready to transfer it to disk. If the data is used by only one process, the buffers are usually local to that process. If other processes will access the data, system memory space and disk I/O operations can be saved by using global buffers to process the data. RMSGBRIO is the number of direct I/Os from disk into the global buffers for the file. If you see RMSGBRIO increasing, it may be an indication that the size of the global buffer is inefficient or the file is not good for buffering. The buffer size may have to be increased to reduce the number of disk I/Os.

R M S G B W I O — File Global Buffer Write I/Os

RMSGBWIO contains the number of global buffer write I/Os to the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSGBRIO - File Global Buffer Read I/Os
RMSLBRIO - File Local Buffer Read I/Os
RMSLBWIO - File Local Buffer Write I/Os

ELEMENT DERIVATION: RMSGBWIO is set to the value of the MNR_RMS$L_GBWDIRIOS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGBWIO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSGBWIO contains the sum of all values of RMSGBWIO processed in the level of summarization.

USAGE NOTES: The data read from or written to an RMS file is kept in buffers until the user process requests it or RMS is ready to transfer it to disk. If the data is used by only one process, the buffers are usually local to that process. If processes will access the data, system memory space and disk I/O operations can be saved by using global buffers to process the data. RMSGBWIO is the number of direct I/Os onto disk from the global buffers for this file.

R M S G C A T T — File Global Cache Attempts

RMSGCATT contains the number of global cache attempts on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSGCHIT - File Global Cache Hits
RMSLCATT - File Local Cache Attempts
RMSLCHIT - File Local Cache Hits

ELEMENT DERIVATION: RMSGCATT is set to the value of the MNR_RMS$L_GCACHEATT field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGCATT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSGCATT contains the sum of all values of RMSGCATT processed in the level of summarization.

USAGE NOTES: RMS indexes are stored in cache buffers to speed up data transfer. For files used by one process, the cache buffers are usually in local storage for that process. For files that are used by multiple processes, much memory space and many disk I/Os can be saved by using global cache buffers. RMSGCATT is the count of times RMS attempted to find an index key in the global cache. RMSGCHIT is the count of times that those attempts were successful.

R M S G C H I T | File Global Cache Hits

RMSGCATT contains the number of successful global cache attempts on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSGCATT - File Global Cache Attempts
RMSLCATT - File Local Cache Attempts
RMSLCHIT - File Local Cache Hits

ELEMENT DERIVATION: RMSGCHIT is set to the value of the MNR_RMS$L_GCACHEHITS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGCHIT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSGCHIT contains the sum of all values of RMSGCHIT processed in the level of summarization.

USAGE NOTES: RMS indexes are stored in cache buffers to speed data transfer. For files used by one process, the cache buffers are usually in local storage for that process. For files that are used by multiple processes, much memory space and many disk I/Os can be saved by using global cache buffers. RMSGCATT is the count of times RMS attempted to find an index key in the global cache. RMSGCHIT is the count of times that those attempts were successful.

R M S G E T B T | File GET Bytes

RMSGETBT contains the number of bytes transferred by $GETs from the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 8 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSGETBT is set to the value of the MNR_RMS$Q_GETBYTES field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGETBT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSGETBT contains the sum of all values of RMSGETBT processed in the level of summarization.

USAGE NOTES: None

R M S G S C N V | File Global Section Conversions

RMSGSCNV contains the number of global section conversions for the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSGSENQ - File Global Section Enqueues
RMSGSDEQ - File Global Section Dequeues

ELEMENT DERIVATION: RMSGSCNV is set to the value of the MNR_RMS$L_GSLCKCNVS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGSCNV takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSGSCNV contains the sum of all values of RMSGSCNV processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed, it then sends a dequeue to free the resource. RMSGSCNV is the number of locks on global sections for this file that were granted.

```
+-----------------+
| R M S G S D E Q |  File Global Section Dequeues
+-----------------+
RMSGSDEQ contains the number of global section dequeues
requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSGSENQ - File Global Section Enqueues
          RMSGSCNV - File Global Section Conversions

ELEMENT DERIVATION: RMSGSDEQ is set to the value of the
MNR_RMS$L_GSLCKDEQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGSDEQ takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: RMSGSDEQ contains the sum of all
values of RMSGSDEQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a
resource in VMS, it issues an enqueue on that resource.  When
the resource becomes available, VMS converts the enqueue into
an exclusive lock for the requesting process.  When the
process has completed, it then sends a dequeue to free the
resource.  RMSGSDEQ is the number of global section locks
freed on this file for DURATION (measurement interval).


+-----------------+
| R M S G S E N Q |  File Global Section Enqueues
+-----------------+
RMSGSENQ contains the number of global section enqueues
requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSGSDEQ - File Global Section Dequeues
          RMSGSCNV - File Global Section Conversions

ELEMENT DERIVATION: RMSGSENQ is set to the value of the
MNR_RMS$L_GSLCKENQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSGSENQ takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: RMSGSENQ contains the sum of all
values of RMSGSENQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a
resource in VMS, it issues an enqueue on that resource.  When
the resource becomes available, VMS converts the enqueue into
an exclusive lock for the requesting process.  When the
process has completed its processing, it then sends a dequeue
to free the resource.  RMSGSENQ is the number of global
section locks requested on this file for DURATION
(measurement interval).


+-----------------+
| R M S K E Y F D |  Keyed Finds
+-----------------+
RMSKEYFD contains the number of keyed $FINDs performed on the
file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSKEYGT - Keyed GETs

ELEMENT DERIVATION: RMSKEYFD is set to the value of the
MNR_RMS$L_KEYFINDS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSKEYFD takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: RMSKEYFD contains the sum of all
values of RMSKEYFD processed in the level of summarization.

USAGE NOTES: A $FIND service with a key in RMS has the
benefit of loading the index caches with the index bucket
that contains the key.  If subsequent $GETs will access that
key and others that are likely to be in that bucket, the
$GETs will be very efficient.


+-----------------+
| R M S K E Y G T |  Keyed GETs
+-----------------+
RMSKEYGT contains the number of keyed $GETs performed on the
file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSKEYFD - Keyed Finds
```

ELEMENT DERIVATION: RMSKEYGT is set to the value of the MNR_RMS$L_KEYGETS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSKEYGT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSKEYGT contains the sum of all values of RMSKEYGT processed in the level of summarization.

USAGE NOTES: When RMS does a keyed $GET, it first executes a keyed $FIND. If the key is in the index cache, the $FIND returns the record file address (RFA) to RMS, which then uses the RFA to bring in the record.

R M S K E Y P T Keyed PUTs

RMSKEYPT contains the number of keyed $PUTs performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSKEYFD - Keyed Finds

ELEMENT DERIVATION: RMSKEYPT is set to the value of the MNR_RMS$L_KEYPUTS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSKEYPT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSKEYPT contains the sum of all values of RMSKEYPT processed in the level of summarization.

USAGE NOTES: When RMS executes a keyed $PUT, it first does a keyed $FIND. If the $FIND locates the key and duplicates are not allowed, an error is returned. Otherwise, the key is inserted in the index and the data written to the file. Inserting the key anywhere but at the end of the index is likely to cause bucket splits.

R M S L B A S T File Local Buffer Blocking ASTs

RMSLBAST contains the number of local buffer blocking asynchronous system traps (ASTs) for the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSFBAST - File Lock Blocking ASTs
RMSGBAST - File Global Buffer Blocking ASTs
RMSLBAST - File Local Buffer Blocking ASTs

ELEMENT DERIVATION: RMSLBAST is set to the value of the MNR_RMS$L_LBLBLKAST field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSLBAST takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSLBAST contains the sum of all values of RMSLBAST processed in the level of summarization.

USAGE NOTES: An AST is a mechanism for signaling when an asynchronous event has been completed. RMSLBAST is a count of such traps used to signal processes that they can now attempt to access the local buffers of the file.

R M S L B C N V File Local Buffer Conversions

RMSLBCNV contains the number of local buffer conversions for the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSLBENQ - File Local Buffer Enqueues
RMSLBDEQ - File Local Buffer Dequeues

ELEMENT DERIVATION: RMSLBCNV is set to the value of the MNR_RMS$L_LBLCKCNVS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSLBCNV takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSLBCNV contains the sum of all values of RMSLBCNV processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed its processing, it then sends a dequeue to free the resource. RMSLBCNV is the number of locks granted to local buffers for this file.

```
+-----------------+
| R M S L B D E Q |  File Local Buffer Dequeues
+-----------------+
RMSLBDEQ contains the number of local buffer dequeues
requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSLBENQ - File Local Buffer Enqueues
          RMSLBCNV - File Local Buffer Conversions

ELEMENT DERIVATION: RMSLBDEQ is set to the value of the
MNR_RMS$L_LBLCKDEQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSLBDEQ takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: RMSLBDEQ contains the sum of all
values of RMSLBDEQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a
resource in VMS, it issues an enqueue on that resource.  When
the resource becomes available, VMS converts the enqueue into
an exclusive lock for the requesting process.  When the
process has completed its processing, it then sends a dequeue
to free the resource.  RMSLBDEQ is the number of local buffer
locks freed on this file for DURATION (measurement interval).


+-----------------+
| R M S L B E N Q |  File Local Buffer Enqueues
+-----------------+
RMSLBENQ contains the number of local buffer enqueues
requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSLBDEQ - File Local Buffer Dequeues
          RMSLBCNV - File Local Buffer Conversions

ELEMENT DERIVATION: RMSLBENQ is set to the value of the
MNR_RMS$L_LBLCKENQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSLBENQ takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: RMSLBENQ contains the sum of all
values of RMSLBENQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a
resource in VMS, it issues an enqueue on that resource.  When
the resource becomes available, VMS converts the enqueue into
an exclusive lock for the requesting process.  When the
process has completed its processing, it then sends a dequeue
to free the resource.  RMSLBENQ is the number of local buffer
locks requested on this file for DURATION (measurement
interval).


+-----------------+
| R M S L B R I O |  File Local Buffer Read I/Os
+-----------------+
RMSLBRIO contains the number of local buffer read I/Os
performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO: RMSGBWIO - File Global Buffer Write I/Os
          RMSGBRIO - File Global Buffer Read I/Os
          RMSLBWIO - File Local Buffer Write I/Os

ELEMENT DERIVATION: RMSLBRIO is set to the value of the
MNR_RMS$L_LBRDIRIOS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSLBRIO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: RMSLBRIO contains the sum of all
values of RMSLBRIO processed in the level of summarization.

USAGE NOTES: The data read from or written to an RMS file is
kept in buffers until the user process requests it or RMS is
ready to transfer it to disk.  If the data is used by only
one process, the buffers are usually local to that process.
If other processes will access the data, system memory space
and disk I/O operations can be saved by using global buffers
to process the data.  RMSLBRIO is the number of direct I/Os
from disk into the local buffers for this file.


+-----------------+
| R M S L B W I O |  File Local Buffer Write I/Os
+-----------------+
RMSLBWIO contains the number of local buffer write I/Os
performed on the file for DURATION (measurement interval).
```

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSGBRIO - File Global Buffer Read I/Os
RMSLBRIO - File Local Buffer Read I/Os
RMSGBWIO - File Global Buffer Write I/Os

ELEMENT DERIVATION: RMSLBWIO is set to the value of the MNR_RMS$L_LBWDIRIOS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSLBWIO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSLBWIO contains the sum of all values of RMSLBWIO processed in the level of summarization.

USAGE NOTES: The data read from or written to an RMS file is kept in buffers until the user process requests it or RMS is ready to transfer it to disk. If the data is used by only one process, the buffers are usually local to that process. If other processes will access the data, system memory space and disk I/O operations can be saved by using global buffers to process the data. RMSLBWIO is the number of direct I/Os onto disk from the local buffers for this file.

R M S L C A T T — File Local Cache Attempts

RMSLCATT contains the number of local cache attempts on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSGCHIT - File Global Cache Hits
RMSGCATT - File Global Cache Attempts
RMSLCHIT - File Local Cache Hits

ELEMENT DERIVATION: RMSLCATT is set to the value of the MNR_RMS$L_LCACHEATT field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSLCATT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSLCATT contains the sum of all values of RMSLCATT processed in the level of summarization.

USAGE NOTES: RMS indexes are stored in cache buffers to speed up data transfer. For files used by one process only, those cache buffers are usually in local storage for that process. For files that are used by multiple processes, much memory space and many disk I/Os can be saved by using global cache buffers. RMSLCATT is the count of times RMS attempted to find an index key in the local cache. RMSLCHIT is the count of times that those attempts were successful.

R M S L C H I T — File Local Cache Hits

RMSLCHIT contains the number of successful local cache attempts on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSGCHIT - File Global Cache Hits
RMSGCATT - File Global Cache Attempts
RMSLCATT - File Local Cache Attempts

ELEMENT DERIVATION: RMSLCHIT is set to the value of the MNR_RMS$L_LCACHEHITS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSLCHIT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSLCHIT contains the sum of all values of RMSLCHIT processed in the level of summarization.

USAGE NOTES: RMS indexes are stored in cache buffers to speed up data transfer. For files used by one process only, those cache buffers are usually in local storage for that process. For files that are used by multiple processes, much memory space and many disk I/Os can be saved by using global cache buffers. RMSLCATT is the count of times RMS attempted to find an index key in the local cache. RMSLCHIT is the count of times that those attempts were successful.

R M S L K C N V — File Lock Conversions

RMSLKCNV contains the number of file lock conversions for the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSLKENQ - File Lock Enqueues
RMSLKDEQ - File Lock Dequeues

ELEMENT DERIVATION: RMSLKCNV is set to the value of the MNR_RMS$L_FLCKCNVS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSLKCNV takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSLKCNV contains the sum of all values of RMSLKCNV processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed its processing, it then sends a dequeue to free the resource. RMSLKCNV is the number of locks granted on the file.

R M S L K D E Q File Lock Dequeues

RMSLKDEQ contains the number of file lock dequeues requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSLKENQ - File Lock Enqueues
RMSLKCNV - File Lock Conversions

ELEMENT DERIVATION: RMSLKDEQ is set to the value of the MNR_RMS$L_FLCKDEQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSLKDEQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSLKDEQ contains the sum of all values of RMSLKDEQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed its processing, it then sends a dequeue to free the resource. RMSLKDEQ is the number of file locks freed on this file for DURATION (measurement interval).

R M S L K E N Q File Lock Enqueues

RMSLKENQ contains the number of file lock enqueues requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSLKDEQ - File Lock Dequeues
RMSLKCNV - File Lock Conversions

ELEMENT DERIVATION: RMSLKENQ is set to the value of the MNR_RMS$L_FLCKENQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSLKENQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSLKENQ contains the sum of all values of RMSLKENQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed, it then sends a dequeue to free the resource. RMSLKENQ is the number of file locks requested on this file for DURATION (measurement interval).

R M S M U L T B File Multibucket Splits

RMSMULTB contains the number of multibucket splits occurred on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSBUCKT - File Bucket Splits

ELEMENT DERIVATION: RMSMULTB is set to the value of the MNR_RMS$L_MBKTSPLT field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSMULTB takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSMULTB contains the sum of all values of RMSMULTB processed in the level of summarization.

USAGE NOTES: Bucket splits occur during insert operations,

when any of the index buffers (buckets) fill up to the level defined in the file header parameter PCTFULL. This happens when keys are inserted between keys in a full index buffer. Split buckets increase file processing time because the process must search for the subsequent indexes in another buffer. It is also likely that when a key search is occurring, the needed bucket will not be in the buffers, causing physical I/Os to continue the search. When buffer space is limited, split buckets often lead to index buffer thrashing, where buckets are read in, overwriting other buckets that will be needed immediately. When the buckets are already split, further splitting forms multibucket splits. This is a sign that the file should be cleaned up with the VMS CONVERT utility.

R M S O P E N | File Opens

RMSOPEN contains the number of times the file was opened for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSOPEN is set to the value of the MNR_RMS$L_OPENS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSOPEN takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSOPEN contains the sum of all values of RMSOPEN processed in the level of summarization.

USAGE NOTES: Opening files is a relatively expensive operation, directories have to be read, the file located, its header read in and the file buffers allocated.

R M S O R G | Organization

RMSORG contains the file organization code for the file.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSORG is set to the value of the MNR_RMS$L_ORG field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSORG takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSORG contains the last value of RMSORG processed in the level of summarization.

USAGE NOTES: RMSORG is an internal code for the organization type of the file. A description of the codes can be found in the VMS Programming Volumes 6A and 6B.

R M S P U T B T | File PUT Bytes

RMSPUTBT contains the number of bytes transferred by $PUT to the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 8 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSPUTBT is set to the value of the MNR_RMS$Q_PUTBYTES field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSPUTBT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSPUTBT contains the sum of all values of RMSPUTBT processed in the level of summarization.

USAGE NOTES: None

R M S R D B T | File Read Bytes

RMSRDBT contains the number of bytes transferred by $READ from the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 8 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSRDBT is set to the value of the

MNR_RMS$Q_READBYTES field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSRDBT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSRDBT contains the sum of all values of RMSRDBT processed in the level of summarization.

USAGE NOTES: None

R M S R E A D | File Reads

RMSREAD contains the number of $READs performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSRDBT - File Read Bytes

ELEMENT DERIVATION: RMSREAD is set to the value of the MNR_RMS$L_READS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSREAD takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSREAD contains the sum of all values of RMSREAD processed in the level of summarization.

USAGE NOTES: $READ operations are for reading RMS files that are either being accessed sequentially or are not in regular record structures.

R M S R E W N D | File Rewinds

RMSREWND contains the number of $REWINDs performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSREWND is set to the value of the MNR_RMS$L_REWINDS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSREWND takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSREWND contains the sum of all values of RMSREWND processed in the level of summarization.

USAGE NOTES: When the file pointer has been moved forward by issuing $READ requests, the user can return the pointer to the starting point by issuing a $REWIND request.

R M S R F A F D | Record File Address Finds

RMSRFAFD contains the number of record file address (RFA) finds performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSRFAFD is set to the value of the MNR_RMS$L_RFAFINDS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSRFAFD takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSRFAFD contains the sum of all values of RMSRFAFD processed in the level of summarization.

USAGE NOTES: The record file address is a relative address within the file structure that allows direct access to the record, regardless of indexes. When a keyed access to a file is performed, RMS uses the RFA returned from a $FIND to give the location of the subsequent $GET or $PUT.

R M S R F A G T | Record File Address GETs

RMSRFAGT contains the number of record file address (RFA) $GETs on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSRFAGT is set to the value of the MNR_RMS$L_RFAGETS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSRFAGT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSRFAGT contains the sum of all values of RMSRFAGT processed in the level of summarization.

USAGE NOTES: An RFA $GET is the fastest way to retrieve data from an indexed file, or from a file that is not record structured. The RFA of information is not necessarily constant, however. RFAs can change because of the use of the VMS CONVERT utility or even a simple BACKUP operation. When RMS does a keyed $GET, it uses a keyed $FIND followed by an RFA $GET.

R M S R L C N V — Record Lock Conversions

RMSRLCNV contains the number of record lock conversions for the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSRLENQ - Record Lock Enqueues
RMSRLDEQ - Record Lock Dequeues

ELEMENT DERIVATION: RMSRLCNV is set to the value of the MNR_RMS$L_RLCKCNVS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSRLCNV takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSRLCNV contains the sum of all values of RMSRLCNV processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed its processing, it then sends a dequeue to free the resource. RMSRLCNV is the number of record locks granted on the file.

R M S R L D E Q — Record Lock Dequeues

RMSRLDEQ contains the number of record lock dequeues requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSRLENQ - Record Lock Enqueues
RMSRLCNV - Record Lock Conversions

ELEMENT DERIVATION: RMSRLDEQ is set to the value of the MNR_RMS$L_RLCKDEQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSRLDEQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSRLDEQ contains the sum of all values of RMSRLDEQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed its processing, it then sends a dequeue to free the resource. RMSRLDEQ is the number of record locks freed on this file for DURATION (measurement interval).

R M S R L E N Q — Record Lock Enqueues

RMSRLENQ contains the number of record lock enqueues requested on the file.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSRLDEQ - Record Lock Dequeues
RMSRLCNV - Record Lock Conversions

ELEMENT DERIVATION: RMSRLENQ is set to the value of the MNR_RMS$L_RLCKENQ field from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSRLENQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSRLENQ contains the sum of all values of RMSRLENQ processed in the level of summarization.

USAGE NOTES: When a process wants exclusive access to a resource in VMS, it issues an enqueue on that resource. When the resource becomes available, VMS converts the enqueue into an exclusive lock for the requesting process. When the process has completed its processing, it then sends a dequeue to free the resource. RMSRLENQ is the number of record locks requested on this file for DURATION (measurement interval).

R M S S E Q F D | Sequential Finds

RMSSEQFD contains the number of sequential $FINDs performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSSEQFD is set to the value of the MNR_RMS$L_SEQFINDS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSSEQFD takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSSEQFD contains the sum of all values of RMSSEQFD processed in the level of summarization.

USAGE NOTES: $FINDs on sequential files only works for files that are record structured.

R M S S E Q G T | Sequential GETs

RMSSEQGT contains the number of sequential $GETs performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSSEQFD - Sequential Finds
RMSSEQPT - Sequential PUTs

ELEMENT DERIVATION: RMSSEQGT is set to the value of the MNR_RMS$L_SEQGETS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSSEQGT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSSEQGT contains the sum of all values of RMSSEQGT processed in the level of summarization.

USAGE NOTES: $GETs to sequential files assumes there is a record structure for the file.

R M S S E Q P T | Sequential PUTs

RMSSEQPT contains the number of sequential $PUTs performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSSEQFD - Sequential Finds
RMSSEQGT - Sequential GETs

ELEMENT DERIVATION: RMSSEQPT is set to the value of the MNR_RMS$L_SEQPUTS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSSEQPT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSSEQPT contains the sum of all values of RMSSEQPT processed in the level of summarization.

USAGE NOTES: $PUTs to sequential files assumes there is a record structure to the file.

R M S T R U B L | File Truncate Blocks

RMSTRUBL contains the number of blocks affected by $TRUNCATEs on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSTRUNC - File Truncates

ELEMENT DERIVATION: RMSTRUBL is set to the value of the MNR_RMS$L_TRUNCBLKS field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSTRUBL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSTRUBL contains the sum of all values of RMSTRUBL processed in the level of summarization.

USAGE NOTES: When a $TRUNCATE is issued, some blocks are given back to the file system. RMSTRUBL counts those blocks for all $TRUNCATE calls on the file.

R M S T R U N C | File Truncates

RMSTRUNC contains the number of $TRUNCATEs on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSTRUBL - File Truncate Blocks

ELEMENT DERIVATION: RMSTRUNC is set to the value of the MNR_RMS$L_TRUNCATES field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSTRUNC takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSTRUNC contains the sum of all values of RMSTRUNC processed in the level of summarization.

USAGE NOTES: The $TRUNCATE operation causes all blocks that are allocated but do not have data written in them to be returned from the file to the file system.

R M S U P D B T | File Update Bytes

RMSUPDBT contains the number of bytes transferred by the $UPDATE on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 8 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSUPDBT is set to value of the MNR_RMS$Q_UPDATEBYTES from the VMS MONITOR RMS record.

RANGE OF VALUE: RMSUPDBT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSUPDBT contains the sum of all values of RMSUPDBT processed in the level of summarization.

USAGE NOTES: None

R M S U P D T E | File Updates

RMSUPDTE contains the number of $UPDATEs performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSUPDBT - File Update Bytes

ELEMENT DERIVATION: RMSUPDTE is set to the value of the MNR_RMS$L_UPDATES field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSUPDTE takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSUPDTE contains the sum of all values of RMSUPDTE processed in the level of summarization.

USAGE NOTES: $UPDATES are only performed to files with a record structure.

R M S W R B T | File Write Bytes

RMSWRBT contains the number of bytes processed by $WRITE on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 8 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: RMSWRBT is set to the value of the MNR_RMS$Q_WRITEBYTES field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSWRBT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSWRBT contains the sum of all

values of RMSWRBT processed in the level of summarization.

USAGE NOTES: None

| R M S W R I T E | File Writes

RMSWRITE contains the number of $WRITEs performed on the file for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: RMSREAD - File Reads

ELEMENT DERIVATION: RMSWRITE is set to the value of the MNR_RMS$L_WRITES field of the VMS MONITOR RMS record.

RANGE OF VALUE: RMSWRITE takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: RMSWRITE contains the sum of all values of RMSWRITE processed in the level of summarization.

USAGE NOTES: The $WRITE operation is used for putting data in a sequential file, especially one that has no particular record structure. $WRITE always puts data at the end of the file. This could cause the system to perform append locking or extension processing.

FILE ABSTRACT: VMS System Profile File	
CONTENT	Quantifies the overall and detailed aspects of system operation in a consolidated file.
USE	To report on a wide variety of measures of system components for performance tuning and capacity planning.
SOURCE	VMS Monitor Class Records for system class, including record types 1, 3, 4, 5, 6, 7, 8, 11, 14, 17, 19, 21, and 22, as documented in the VMS Monitor Reference Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	YEAR	MONTH	DAY	HOUR
	ENDTS				
DAYS	SYSID	YEAR	MONTH	DAY	HOUR
WEEKS	SYSID	YEAR	WEEK	ZONE	HOUR
MONTHS	SYSID	YEAR	MONTH	ZONE	
YEARS	N/A				
TABLES	N/A				

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
	Sequence/Summary Data Elements	
XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
XDWM..E	ZONE	- Time Zone
	Common Data Elements	
XD....E	DAYNAME	- Name of Day of Week
X.....E	DEXMONVR	- VMS Monitor Version
XDWM..E	DURATION	- Recording Interval Time
XDWM..E	ENDTS	- End Time Stamp
XDWM..E	INTERVLS	- Number of Recording Intervals
X.....E	ORGSYSID	- Originating System Identification

```
XDWM..E  STARTTS  - Start Time Stamp

Retained Data Elements

X.....   SPRBIGHL - Largest Block Unused
X.....   SPRCEF   - Common Event Flag Wait
X.....   SPRCOLPG - Collided Page Wait
X.....E  SPRCOM   - Compute State Wait, Inswapped
X.....E  SPRCOMO  - Compute State Wait, Outswapped
X.....E  SPRCUR   - Current Processes
X.....E  SPRCURLK - Current Locks
X.....E  SPRCURRS - Current Resources
X.....   SPRFPGWT - Free Page Wait
X.....E  SPRFRE   - Free Page List Size
X.....E  SPRFREEP - IO Free Page Count
X.....   SPRFRPG  - Free Page Count
X.....E  SPRFRPGC - PGE Free Page Count
X.....   SPRHIB   - Hibernate Wait, Inswapped
X.....   SPRHIBO  - Hibernate Wait, Outswapped
X.....   SPRHOLES - Nonpaged Pool Unused KBytes
X.....E  SPRIPFRE - Intermediate Request Packets Free
X.....E  SPRIPUSE - Intermediate Request Packets in Use
X.....   SPRKBUSE - Nonpaged Pool KBytes in Use
X.....   SPRLEF   - Local Event Flag Wait, Inswapped
X.....   SPRLEFO  - Local Event Flag Wait, Outswapped
X.....E  SPRLPFRE - Large Request Packets Free
X.....E  SPRLPUSE - Large Request Packets in Use
X.....E  SPRLRGPK - Large Request Packets Left
X.....E  SPRMASK  - Record Construction Audit Mask
X.....   SPRMDPG  - Modified Page Count
X.....E  SPRMDPGC - PGE Modified Page Count
X.....E  SPRMODP  - IO Modified Page Count
X.....   SPRMWAIT - Miscellaneous Resource Wait
X.....   SPROSTAT - Other States
X.....   SPRPFW   - Page Fault Wait
X.....E  SPRPRCNT - Process Count
X.....   SPRSMBLK - Block Less Than or Equal to 32
X.....   SPRSMHL  - Smallest Block Unused
X.....E  SPRSPFRE - Small Request Packets Free
X.....E  SPRSPUSE - Small Request Packets in Use
X.....   SPRSUSP  - Suspended Wait, Inswapped
X.....   SPRSUSPO - Suspended Wait, Outswapped
X.....   SPRUNCON - Nonpaged Pool Unused Contiguous Space

Accumulated Data Elements

XDWM..   SPRACCS  - File Access
XDWM..E  SPRALLOC - Disk Allocations
XDWM..E  SPRARLPK - Arriving Local Packets
XDWM..E  SPRARTPK - Arriving Transit Packets
XDWM..   SPRBAI   - Blocking ASTs, Incoming
XDWM..   SPRBAL   - Blocking ASTs, Local
XDWM..   SPRBAO   - Blocking ASTs, Outgoing
XDWM..E  SPRBIOS  - Buffered I/Os
XDWM..   SPRBITAT - Storage Bitmap Cache Attempts
XDWM..   SPRBITHT - Storage Bitmap Cache Hits
XDWM..E  SPRBLAST - Blocking ASTs
XDWM..   SPRBUFFL - Receiver Buffer Failures
XDWM..E  SPRBUFIO - Buffered I/Os
XDWM..E  SPRBWAIT - MSCP Buffer Waits
XDWM..E  SPRCALLS - FCP Calls
XDWM..E  SPRCENQ  - Converted Enqueues
XDWM..E  SPRCLSCT - CLS CLUSTER Record Count (Type 19)
XDWM..E  SPRCLSDR - CLS CLUSTER Record Duration (Type 19)
XDWM..E  SPRCLSTM - CLS CPU Busy Time
XDWM..E  SPRCNTAB - Transactions Aborted Count
XDWM..E  SPRCNTAD - Add Remote Branch Ops Count
XDWM..E  SPRCNTBR - Start Remote Branch Ops Count
XDWM..E  SPRCNTCO - Transactions Committed Count
XDWM..   SPRCNTD1 - Transactions < 1 Second Count
XDWM..   SPRCNTD2 - Transactions 1-2 Seconds Count
XDWM..   SPRCNTD3 - Transactions 2-3 Seconds Count
XDWM..   SPRCNTD4 - Transactions 3-4 Seconds Count
XDWM..   SPRCNTD5 - Transactions 4-5 Seconds Count
XDWM..   SPRCNTD6 - Transactions > 5 Seconds Count
XDWM..E  SPRCNTEN - Transactions Ended Count
XDWM..E  SPRCNTOP - One-Phase Commits Initiated
XDWM..E  SPRCNTPR - Transactions Prepared Count
XDWM..E  SPRCNTST - Transactions Started Count
XDWM..E  SPRCPUNI - Number of Instructions Executed
XDWM..E  SPRCPUTM - System CPU Busy Time
XDWM..   SPRDATAT - Directory Data Cache Attempts
XDWM..   SPRDATHT - Directory Data Cache Hits
XDWM..E  SPRDEADF - Deadlocks Found
XDWM..E  SPRDEADS - Deadlock Searches
XDWM..E  SPRDEQ   - Dequeues
XDWM..E  SPRDIOS  - Direct I/Os
XDWM..E  SPRDIRIO - Direct I/Os
XDWM..E  SPRDLKCT - DLK DLOCK Record Count (Type 14)
XDWM..E  SPRDLKDR - DLK DLOCK Record Duration (Type 14)
XDWM..   SPRDLM   - Deadlock Messages
XDWM..E  SPRDMZFL - Demand Zero Page Faults
XDWM..E  SPRDNTCT - DNT DECNET Record Count (Type 8)
XDWM..E  SPRDNTDR - DNT DECNET Record Duration (Type 8)
XDWM..E  SPRDPLPK - Departing Local Packets
XDWM..   SPRDQI   - Dequeue Incoming
XDWM..   SPRDQL   - Dequeue Local
XDWM..   SPRDQO   - Dequeue Outgoing
XDWM..   SPRECI   - Enqueue Conversions Incoming
XDWM..   SPRECL   - Enqueue Conversions Local
XDWM..   SPRECO   - Enqueue Conversions Outgoing
XDWM..E  SPRENQNQ - Enqueues Not Queued
XDWM..E  SPRENQWT - Enqueue Waits
XDWM..E  SPRERASE - Erase I/O Operations
```

```
XDWM..   SPREXTAT - Extent Cache Attempts
XDWM..   SPREXTHT - Extent Cache Hits
XDWM..   SPRFCBAT - Directory FCB Cache Attempts
XDWM..   SPRFCBHT - Directory FCB Cache Hits
XDWM..E  SPRFCPCT - FCP Record Count (Type 5)
XDWM..E  SPRFCPDR - FCP Record Duration (Type 5)
XDWM..   SPRFCPFL - FCP Page Faults
XDWM..E  SPRFCPRI - FCP Disk Read I/Os
XDWM..E  SPRFCPTM - FCP CPU Time
XDWM..E  SPRFCPWI - FCP Disk Write I/Os
XDWM..   SPRFHDAT - File Header Cache Attempts
XDWM..   SPRFHDHT - File Header Cache Hits
XDWM..   SPRFIDAT - File ID Cache Attempts
XDWM..   SPRFIDHT - File ID Cache Hits
XDWM..E  SPRFILOP - Files Opened
XDWM..E  SPRFNI   - Directory Functions Incoming
XDWM..E  SPRFNO   - Directory Functions Outgoing
XDWM..E  SPRFPGFL - Free List Page Faults
XDWM..E  SPRFRGMT - MSCP Fragments
XDWM..E  SPRFSCCT - FSC CACHE Record Count (Type 11)
XDWM..E  SPRFSCDR - FSC CACHE Record Duration (Type 11)
XDWM..E  SPRGBLFL - Global Valid Page Faults
XDWM..E  SPRINSWP - Inswaps
XDWM..E  SPRIOVCT - IOV IO Record Count (Type 4)
XDWM..E  SPRIOVDR - IOV IO Record Duration (Type 4)
XDWM..E  SPRIOVFL - IO Page Faults
XDWM..   SPRLCI   - Lock Conversions, Incoming
XDWM..   SPRLCL   - Lock Conversions, Local
XDWM..   SPRLCO   - Lock Conversions, Outgoing
XDWM..E  SPRLOGNM - Logical Name Translations
XDWM..E  SPRLOK   - Total All Locks
XDWM..E  SPRLOKCT - LOK LOCK Record Count (Type 7)
XDWM..E  SPRLOKDR - LOK LOCK Record Duration (Type 7)
XDWM..   SPRMAILW - Mailbox Writes
XDWM..E  SPRMPGFL - Modified List Page Faults
XDWM..E  SPRMSCCT - MSC MSCP Record Count (Type 21)
XDWM..E  SPRMSCDR - MSC MSCP Record Duration (Type 21)
XDWM..E  SPRMSCRD - MSCP Reads
XDWM..E  SPRMSCSP - MSCP Splits
XDWM..E  SPRMSCWR - MSCP Writes
XDWM..   SPRNEI   - New Enqueue Incoming
XDWM..   SPRNEL   - New Enqueue Local
XDWM..E  SPRNENQ  - New Enqueues
XDWM..   SPRNEO   - New Enqueue Outgoing
XDWM..E  SPRNEWFL - New Files
XDWM..   SPRNLI   - New Locks, Incoming
XDWM..   SPRNLL   - New Locks, Local
XDWM..   SPRNLO   - New Locks, Outgoing
XDWM..E  SPROPEN  - Files Opened
XDWM..E  SPRPGECT - PGE PAGE Record Count (Type 3)
XDWM..E  SPRPGEDR - PGE PAGE Record Duration (Type 3)
XDWM..E  SPRPGEFL - PGE Page Faults
XDWM..E  SPRPGERD - PGE Page Reads
XDWM..E  SPRPGERI - PGE Page Read I/Os
XDWM..E  SPRPGEWI - PGE Page Write I/Os
XDWM..E  SPRPGEWR - PGE Page Writes
XDWM..E  SPRPGFLT - Page Faults
XDWM..E  SPRPGRD  - Page Reads
XDWM..E  SPRPGWT  - Page Writes
XDWM..   SPRPKLOS - Transit Packets Lost
XDWM..E  SPRPOLCT - POL POOL Record Count (Type 6)
XDWM..E  SPRPOLDR - POL POOL Record Duration (Type 6)
XDWM..E  SPRPRDIO - Page Read I/Os
XDWM..E  SPRPWIO  - Page Write I/Os
XDWM..   SPRQUOAT - Quota Cache Attempts
XDWM..   SPRQUOHT - Quota Cache Hits
XDWM..E  SPRRDIOS - Page Read I/Os
XDWM..E  SPRREQST - MSCP Requests
XDWM..   SPRSIZE1 - 1 Block I/Os
XDWM..   SPRSIZE2 - 2-3 Block I/Os
XDWM..   SPRSIZE3 - 4-7 Block I/Os
XDWM..   SPRSIZE4 - 8-15 Block I/Os
XDWM..   SPRSIZE5 - 16-31 Block I/Os
XDWM..   SPRSIZE6 - 32-63 Block I/Os
XDWM..   SPRSIZE7 - 64 And Larger Block I/Os
XDWM..   SPRSPLIT - Split Transfers
XDWM..E  SPRSTACT - STA STATES Record Count (Type 1)
XDWM..E  SPRSTADR - STA STATES Record Duration (Type 1)
XDWM..E  SPRSYSFL - System Page Faults
XDWM..E  SPRSYTCT - SYT SYSTEM Record Count (Type 17)
XDWM..E  SPRSYTDR - SYT SYSTEM Record Duration (Type 17)
XDWM..E  SPRTRNCT - TRN TRANS Record Count (Type 22)
XDWM..E  SPRTRNDR - TRN TRANS Record Duration (Type 22)
XDWM..   SPRTURN  - Window Turns
XDWM..   SPRUNI   - Unlocks, Incoming
XDWM..   SPRUNL   - Unlocks, Local
XDWM..   SPRUNO   - Unlocks, Outgoing
XDWM..   SPRVLOK  - Volume Lock Waits
XDWM..E  SPRWIPFL - Write In Progress Page Faults
XDWM..   SPRWTURN - Window Turns
```

Minimum Data Elements

```
XDWM..   SPRMNFP  - Minimum Free Page Count
XDWM..E  SPRMNFRE - Minimum Free Page List Size
XDWM..   SPRMNHOL - Minimum Unused KBytes
XDWM..E  SPRMNIFP - IO Minimum Free Page Count
XDWM..E  SPRMNIMP - IO Minimum Modified Page Count
XDWM..E  SPRMNIPF - Minimum Intermediate Req Packets Free
XDWM..   SPRMNLBK - Minimum Largest Block
XDWM..E  SPRMNLPF - Minimum Large Request Packets Free
XDWM..E  SPRMNLPK - Minimum Large Request Packets Left
XDWM..   SPRMNMP  - Minimum Modified Page Count
XDWM..E  SPRMNPFP - PGE Minimum Free Page Count
```

```
| XDWM..E  SPRMNPMP - PGE Minimum Modified Page Count
| XDWM..   SPRMNSBK - Minimum Smallest Block
| XDWM..E  SPRMNSPF - Minimum Small Request Packets Free
| XDWM..   SPRMNUCN - Minimum Unused Contiguous Space
|
| Maximum Data Elements
|
| XDWM..   SPRMSTAB - Max Transaction Abort Rate
| XDWM..   SPRMSTAD - Max Transaction Remote Add Rate
| XDWM..   SPRMSTBR - Max Transaction Remote Start Rate
| XDWM..   SPRMSTCO - Max Transaction Total Commit Rate
| XDWM..   SPRMSTEN - Max Transaction End Rate
| XDWM..   SPRMSTOP - Max Transaction One-Phase Init Rate
| XDWM..   SPRMSTPR - Max Transaction Prepare Rate
| XDWM..   SPRMSTST - Max Transaction Start Rate
| XDWM..   SPRMXCEF - Maximum Common Event Flag Wait
| XDWM..E  SPRMXCLK - Maximum Current Locks
| XDWM..E  SPRMXCP  - Maximum Current Processes
| XDWM..   SPRMXCPW - Maximum Collided Page Wait
| XDWM..E  SPRMXCRS - Maximum Current Resources
| XDWM..E  SPRMXCSI - Maximum Compute State Wait, Inswapped
| XDWM..E  SPRMXCSO - Maximum Compute State Wait, Outswapped
| XDWM..   SPRMXFPW - Maximum Free Page Wait
| XDWM..   SPRMXHWI - Maximum Hibernate Wait, Inswapped
| XDWM..   SPRMXHWO - Maximum Hibernate Wait, Outswapped
| XDWM..E  SPRMXIPU - Maximum Intermediate Req Packet in Use
| XDWM..   SPRMXKBU - Maximum Nonpaged Pool KBytes in Use
| XDWM..   SPRMXLEI - Maximum Local Event Flag Wait, Inswap
| XDWM..   SPRMXLEO - Maximum Local Event Flag Wait, Outswap
| XDWM..E  SPRMXLPU - Maximum Large Request Packets in Use
| XDWM..   SPRMXMRW - Maximum Miscellaneous Resource Wait
| XDWM..E  SPRMXOST - Maximum Other States
| XDWM..   SPRMXPFW - Maximum Page Fault Wait
| XDWM..E  SPRMXPRC - Maximum Process Count
| XDWM..   SPRMXSMB - Maximum Block Less Than or Equal to 32
| XDWM..   SPRMXSPI - Maximum Suspended Wait, Inswapped
| XDWM..   SPRMXSPO - Maximum Suspended Wait, Outswapped
| XDWM..E  SPRMXSPU - Maximum Small Request Packets in Use
| XDWM..E  SPRTRNGE - DEXTS Range From Input Records
|
| Derived Data Elements
|
| XDWM..   SPRPCBIT - Storage Bitmap Cache Hit Percentage
| XDWM..   SPRPCDDC - Directory Data Cache Hit Percentage
| XDWM..   SPRPCEXT - Extent Cache Hit Percentage
| XDWM..   SPRPCFCB - Directory FCB Cache Hit Percentage
| XDWM..   SPRPCFHC - File Header Cache Hit Percentage
| XDWM..   SPRPCFID - File ID Cache Hit Percentage
| XDWM..   SPRPCQOT - Quota Cache Hit Percentage
| XDWM..   SPRPSACC - File Access Rate
| XDWM..E  SPRPSALL - Disk Allocation Rate
| XDWM..E  SPRPSARL - Arriving Local Packet Rate
| XDWM..E  SPRPSAST - Blocking ASTs Rate
| XDWM..E  SPRPSATP - Arriving Transit Packet Rate
| XDWM..   SPRPSBAI - Blocking ASTs Incoming Rate
| XDWM..   SPRPSBAL - Blocking ASTs Local Rate
| XDWM..   SPRPSBAO - Blocking ASTs Outgoing Rate
| XDWM..E  SPRPSBIO - Buffered I/O Rate
| XDWM..   SPRPSBIT - Storage Bitmap Cache Attempt Rate
| XDWM..E  SPRPSBUF - Buffered I/O Rate
| XDWM..E  SPRPSCAL - FCP Call Rate
| XDWM..E  SPRPSCEQ - Converted Enqueue Rate
| XDWM..   SPRPSDDC - Directory Data Cache Attempt Rate
| XDWM..E  SPRPSDEQ - Dequeue Rate
| XDWM..E  SPRPSDFD - Deadlocks Found Rate
| XDWM..E  SPRPSDIO - Direct I/O Rate
| XDWM..E  SPRPSDIR - Direct I/O Rate
| XDWM..   SPRPSDLM - Deadlock Message Rate
| XDWM..E  SPRPSDMZ - Demand Zero Page Fault Rate
| XDWM..E  SPRPSDPL - Departing Local Packet Rate
| XDWM..   SPRPSDQI - Dequeue Incoming Rate
| XDWM..   SPRPSDQL - Dequeue Local Rate
| XDWM..   SPRPSDQO - Dequeue Outgoing Rate
| XDWM..E  SPRPSDSR - Deadlock Search Rate
| XDWM..   SPRPSECI - Enqueue Conversion Incoming Rate
| XDWM..   SPRPSECL - Enqueue Conversion Local Rate
| XDWM..   SPRPSECO - Enqueue Conversion Outgoing Rate
| XDWM..E  SPRPSEIO - Erase I/O Rate
| XDWM..E  SPRPSENQ - Enqueues Not Queued Rate
| XDWM..E  SPRPSEWT - Enqueue Waits Rate
| XDWM..   SPRPSEXT - Extent Cache Attempt Rate
| XDWM..   SPRPSFCB - Directory FCB Cache Attempt Rate
| XDWM..   SPRPSFFL - FCP Page Fault Rate
| XDWM..   SPRPSFHC - File Header Cache Attempt Rate
| XDWM..   SPRPSFID - File ID Cache Attempt Rate
| XDWM..E  SPRPSFLT - Page Fault Rate
| XDWM..E  SPRPSFNI - Directory Functions Incoming Rate
| XDWM..E  SPRPSFNO - Directory Functions Outgoing Rate
| XDWM..E  SPRPSFOP - Files Opened Rate
| XDWM..E  SPRPSFPL - Free List Page Fault Rate
| XDWM..E  SPRPSFRI - FCP Disk Read I/O Rate
| XDWM..E  SPRPSFWI - FCP Disk Write I/O Rate
| XDWM..E  SPRPSGBL - Global Valid Page Fault Rate
| XDWM..E  SPRPSIFL - IO Page Fault Rate
| XDWM..E  SPRPSINS - Inswap Rate
| XDWM..   SPRPSLCI - Lock Conversion Incoming Rate
| XDWM..   SPRPSLCL - Lock Conversion Local Rate
| XDWM..   SPRPSLCO - Lock Conversion Outgoing Rate
| XDWM..E  SPRPSLNT - Logical Name Translation Rate
| XDWM..   SPRPSLOS - Transit Packet Loss Rate
| XDWM..   SPRPSMBX - Mailbox Write Rate
| XDWM..E  SPRPSMPL - Modified List Page Fault Rate
| XDWM..   SPRPSNEI - New Enqueue Incoming Rate
| XDWM..   SPRPSNEL - New Enqueue Local Rate
|
```

```
XDWM..   SPRPSNEO - New Enqueue Outgoing Rate
XDWM..E  SPRPSNEQ - New Enqueue Rate
XDWM..E  SPRPSNFL - New Files Rate
XDWM..   SPRPSNLI - New Locks Incoming Rate
XDWM..   SPRPSNLL - New Locks Local Rate
XDWM..   SPRPSNLO - New Locks Outgoing Rate
XDWM..E  SPRPSOPN - File Open Rate
XDWM..E  SPRPSPFL - PGE Page Fault Rate
XDWM..E  SPRPSPGR - Page Read Rate
XDWM..E  SPRPSPGW - Page Write Rate
XDWM..E  SPRPSPR  - Page Read I/O Rate
XDWM..E  SPRPSPRI - PGE Page Read I/O Rate
XDWM..E  SPRPSPW  - Page Write I/O Rate
XDWM..E  SPRPSPWI - PGE Page Write I/O Rate
XDWM..   SPRPSQOT - Quota Cache Attempt Rate
XDWM..   SPRPSRBF - Receiver Buffer Failure Rate
XDWM..E  SPRPSRED - PGE Page Read Rate
XDWM..E  SPRPSRIO - Read I/O Rate
XDWM..   SPRPSSPL - Split Transfer Rate
XDWM..E  SPRPSSYS - System Page Fault Rate
XDWM..E  SPRPSTAB - Transaction Abort Rate
XDWM..E  SPRPSTAD - Transaction Remote Add Rate
XDWM..E  SPRPSTBR - Transaction Remote Start Rate
XDWM..E  SPRPSTCO - Transaction Total Commit Rate
XDWM..E  SPRPSTEN - Transaction End Rate
XDWM..E  SPRPSTOP - Transaction One-Phase Init Rate
XDWM..E  SPRPSTPR - Transaction Prepare Rate
XDWM..   SPRPSTRN - Window Turn Rate
XDWM..E  SPRPSTST - Transaction Start Rate
XDWM..   SPRPSUNI - Unlock Incoming Rate
XDWM..   SPRPSUNL - Unlock Local Rate
XDWM..   SPRPSUNO - Unlock Outgoing Rate
XDWM..   SPRPSVLK - Volume Lock Wait Rate
XDWM..E  SPRPSWIP - Write In Progress Page Fault Rate
XDWM..E  SPRPSWRT - PGE Page Write Rate
```

```
+-----------------+
| S P R A C C S   |  File Access
+-----------------+
SPRACCS contains the number of directory accesses performed
to locate files during DURATION (measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSACC - File Access Rate
           SPRNEWFL - New Files
           SPRTURN  - Window Turns
           SPROPEN  - Files Opened
           SPRERASE - Erase I/O Operations

ELEMENT DERIVATION:  SPRACCS is set to the value of the
MNR_FCP$L_ACCESS field of the VMS MONITOR FCP record.

RANGE OF VALUE:  SPRACCS takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRACCS contains the sum of all
values of SPRACCS processed in the level of summarization.

USAGE NOTES:  File accesses are directory lookups for the
location of the file.  SPRACCS is incremented whenever a file
is created, extended, opened, deleted, or erased and when a
window turn occurs.  Directory lookups are only costly when
the file system cache is not large enough for the directories
regularly required on the system.  If the SPRACCS is large,
the SYSGEN parameters ACP_SYSACC, ACP_DINDXCACHE,
ACP_DIRCACHE, and ACP_MAPCACHE may need adjusting to improve
performance.


+-----------------+
| S P R A L L O C |  Disk Allocations
+-----------------+
SPRALLOC contains the number of queued I/O requests that
caused disk space to be allocated during DURATION
(measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSALL - Disk Allocation Rate
           SPRNEWFL - New Files
           SPRFCPWI - FCP Disk Write I/Os

ELEMENT DERIVATION:  SPRALLOC is set to the value of the
MNR_FCP$L_ALLOC field of the VMS MONITOR FCP record.

RANGE OF VALUE:  SPRALLOC takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRALLOC contains the sum of all
values of SPRALLOC processed in the level of summarization.

USAGE NOTES:  Space is allocated when a file is created or
when a write request causes the file to be extended.  If
SPRALLOC is high, then the file system should be checked for
fragmentation.  Fragmentation can severely affect disk I/O
performance as well as overall system performance.  Disk I/O
performance is affected by the extra seek time moving from
one portion of a file to another.  System performance can be
```

degraded if the fragmentation causes a high rate of window turns.

| S P R A R L P K | Arriving Local Packets

SPRARLPK contains the number of arriving local packets.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSARL - Arriving Local Packet Rate

ELEMENT DERIVATION: SPRARLPK is set to the value of the MNR_NET$L_ARRLOCPK field of the VMS MONITOR DECNET record.

RANGE OF VALUE: SPRARLPK takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRARLPK contains the sum of all values of SPRARLPK processed in the level of summarization.

USAGE NOTES: SPRARLPK counts the packets received by this node that were addressed here. These packets are not just limited to data packets but include sequencing and other control packets.

| S P R A R T P K | Arriving Transit Packets

SPRARTPK contains the number of arriving transit packets.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSATP - Arriving Transit Packet Rate

ELEMENT DERIVATION: SPRARTPK is set to the value of the MNR_NET$L_ARRTRAPK field of the VMS MONITOR DECNET record.

RANGE OF VALUE: SPRARTPK takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRARTPK contains the sum of all values of SPRARTPK processed in the level of summarization.

USAGE NOTES: Transit packets are those that arrive at this node to be routed on to a subsequent node. If this node is a routing node, the ratio between SPRARTPK and SPRARLPK gives a good estimate of the effect of the routing function on the node.

| S P R B A I | Blocking ASTs, Incoming

SPRBAI contains the count of incoming blocking ASTs for the Lock Manager.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRBAL - Blocking ASTs, Local
SPRBAO - Blocking ASTs, Outgoing
SPRBLAST - Blocking ASTs

ELEMENT DERIVATION: SPRBAI is set to the value of the MNR_DLO$L_BLKIN field in the VMS MONITOR DLOCK record.

RANGE OF VALUE: SPRBAI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRBAI contains the sum of all values of SPRBAI processed in the level of summarization.

USAGE NOTES: SPRBAI is the number of requests for asynchronous system traps to notify processes on remote nodes that their requests are being served.

| S P R B A L | Blocking ASTs, Local

SPRBAL contains the count of local blocking ASTs for the Lock Manager.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRBAI - Blocking ASTs, Incoming
SPRBAO - Blocking ASTs, Outgoing
SPRBLAST - Blocking ASTs

ELEMENT DERIVATION: SPRBAL is set to the value of the MNR_DLO$L_BLKLOC field in the VMS MONITOR DLOCK record.

RANGE OF VALUE: SPRBAL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRBAL contains the sum of all values of SPRBAL processed in the level of summarization.

USAGE NOTES: SPRBAL is the number of requests for asynchronous system traps to notify local processes that their requests are being served.

| S P R B A O | Blocking ASTs, Outgoing

SPRBAO contains the count of outgoing blocking ASTs for the Lock Manager.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRBAL - Blocking ASTs, Local
SPRBAI - Blocking ASTs, Incoming
SPRBLAST - Blocking ASTs

ELEMENT DERIVATION: SPRBAO is set to the value of the MNR_DLO$L_BLKOUT field in the VMS MONITOR DLOCK record.

RANGE OF VALUE: SPRBAO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRBAO contains the sum of all values of SPRBAO processed in the level of summarization.

USAGE NOTES: SPRBAO is the number of requests for asynchronous system traps on remote nodes to notify processes on this node that their requests are being served.

| S P R B I G H L | Largest Block Unused

SPRBIGHL is the size (in kilobytes) of the largest block of unused space in the dynamic nonpaged pool.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMNLBK - Minimum Largest Block

ELEMENT DERIVATION: SPRBIGHL is set to the value of the MNR_POO$L_BIGHOLE field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRBIGHL takes any integer value from 0 to 2,097,152.

SUMMARIZATION PROCESS: SPRBIGHL contains the last value of SPRBIGHL processed in the level of summarization.

USAGE NOTES: As the nonpaged pool is used, areas that were used and then freed are called holes. The size of the largest hole is stored in SPRBIGHL. When dynamic memory is required, if the size is larger than SPRBIGHL, the memory manager knows that it cannot serve that much contiguous space to the process. If SPRBIGHL is too small, the dynamic portion of the pool is being fragmented badly. If this is the case, the applications should be reviewed for their use of the pool.

| S P R B I O S | Buffered I/Os

SPRBIOS contains the number of I/O requests to buffered devices such as terminals and printers.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROBUFIO - Buffered I/Os
SPRBUFIO - Buffered I/Os
PRCBUFIO - Buffered I/Os
IMGBUFIO - Buffered I/Os

ELEMENT DERIVATION: SPRBIOS is set to the value of the MNR_SYS$L_BUFIO field of the VMS MONITOR I/O record.

RANGE OF VALUE: SPRBIOS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRBIOS contains the sum of all values of SPRBIOS processed in the level of summarization.

USAGE NOTES: SPRBIOS reflects the number of buffered I/O operations, not the size of those operations. Buffered I/O operations usually do not affect system performance much. Systems with all terminals, printers, and other buffered devices under direct control of the CPU, however, can be affected by large numbers of buffered I/Os. If buffered I/O is a problem on the system, reconfiguring hardware to include terminal servers will help reduce the effect on the system.

| S P R B I T A T | Storage Bitmap Cache Attempts

SPRBITAT contains the number of attempts to find information in the storage bitmap cache.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRBITHT - Storage Bitmap Cache Hits

ELEMENT DERIVATION: SPRBITAT is set to the value of the MNR_FIL$L_STORAGMAP_TRIES field of the VMS MONITOR FILE_SYSTEM_CACHE record.

RANGE OF VALUE: SPRBITAT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRBITAT contains the sum of all values of SPRBITAT processed in the level of summarization.

USAGE NOTES: When space is allocated for a file, the first place searched is the extent cache. If there is not enough space there, the storage bitmap is tried. The storage bitmap cache is checked for an area large enough for the file. This check is registered in SPRBITAT. If the space is found in the cache, a hit is also recorded in SPRBITHT. Activity on the storage bitmap is an indication that the disk volume is fragmented. If SPRBITAT is much larger than SPRBITHT, the disk is very badly fragmented and should be defragmented. The size of the bitmap cache is set by the SYSGEN parameter ACP_MAPCACHE.

| S P R B I T H T | Storage Bitmap Cache Hits

SPRBITHT contains the number of successful attempts to find information in the storage bitmap cache.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRBITAT - Storage Bitmap Cache Attempts

ELEMENT DERIVATION: SPRBITHT is set to the value of the MNR_FIL$L_STORAGMAP_HIT field of the VMS MONITOR FILE_SYSTEM_CACHE record.

RANGE OF VALUE: SPRBITHT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRBITHT contains the sum of all values of SPRBITHT processed in the level of summarization.

USAGE NOTES: When space is allocated for a file, the first place searched is the extent cache. If there is not enough space there, the storage bitmap is tried. The storage bitmap cache is checked for an area large enough for the file. This check is registered in SPRBITAT. If the space is found in the cache, a hit is also recorded in SPRBITHT. Activity on the storage bitmap is an indication that the disk volume is fragmented. If SPRBITAT is much larger than SPRBITHT, the disk is very badly fragmented and should be defragmented. The size of the bitmap cache is set by the SYSGEN parameter ACP_MAPCACHE.

| S P R B L A S T | Blocking ASTs

SPRBLAST contains the count of blocking ASTs for the Lock Manager.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSAST - Blocking ASTs Rate
SPRBAL - Blocking ASTs, Local

ELEMENT DERIVATION: SPRBLAST is set to the value of the MNR_LCK$L_BLKAST field of the VMS MONITOR LOCK record.

RANGE OF VALUE: SPRBLAST takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRBLAST contains the sum of all values of SPRBLAST processed in the level of summarization.

USAGE NOTES: SPRBLAST is the number of requests for asynchronous system traps to notify local processes that their requests are being served.

| S P R B U F F L | Receiver Buffer Failures

SPRBUFFL contains the number of receiver buffer failures.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSRBF - Receiver Buffer Failure Rate

ELEMENT DERIVATION: SPRBUFFL is set to the value of the MNR_NET$L_RCVBUFFL field of the VMS MONITOR DECNET record.

RANGE OF VALUE: SPRBUFFL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRBUFFL contains the sum of all values of SPRBUFFL processed in the level of summarization.

USAGE NOTES: When the target node was unable to receive a packet, this node knows it must retransmit the message. SPRBUFFL represents packets that were resent because of buffer problems in the target nodes. If this occurs, the target node is having problems with its nonpaged pool.

S P R B U F I O — Buffered I/Os

SPRBUFIO contains the number of I/O requests to buffered devices, such as terminals and printers.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROBUFIO - Buffered I/Os
SPRBIOS - Buffered I/Os
PRCBUFIO - Buffered I/Os
IMGBUFIO - Buffered I/Os

ELEMENT DERIVATION: SPRBUFIO is set to the value of the MNR_IO$L_BUFIO field of the VMS MONITOR IO record.

RANGE OF VALUE: SPRBUFIO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRBUFIO contains the sum of all values of SPRBUFIO processed in the level of summarization.

USAGE NOTES: SPRBUFIO reflects the number of buffered I/O operations, not the size of those operations. Buffered I/O operations usually do not affect system performance much. Systems with all terminals, printers, and other buffered devices under direct control of the CPU, however, can be affected by large numbers of buffered I/Os. If buffered I/O is a problem on the system, reconfiguring hardware to include terminal servers will help reduce the impact to the system.

S P R B W A I T — MSCP Buffer Waits

SPRBWAIT contains the count of requests that were forced to wait for a buffer to be served.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SPRBWAIT is set to the value of the MNR_MSC$L_BUFWAIT field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRBWAIT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRBWAIT contains the sum of all values of SPRBWAIT processed in the level of summarization.

USAGE NOTES: If there is a high number of requests that are waiting for buffers to be clear, the performance of the cluster will suffer. More MSCP buffer space is needed if SPRBWAIT becomes large.

S P R C A L L S — FCP Calls

SPRCALLS contains the number of queued I/O requests received by the file system during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSCAL - FCP Call Rate

ELEMENT DERIVATION: SPRCALLS is set to the value of the MNR_FCP$L_CALLS field of the VMS MONITOR FCP record.

RANGE OF VALUE: SPRCALLS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRCALLS contains the sum of all values of SPRCALLS processed in the level of summarization.

USAGE NOTES: SPRCALLS is an indicator of the activity on the file system that is due to typical application programs.

| S P R C E F | Common Event Flag Wait

SPRCEF contains the count of processes waiting for a common event flag to be signaled.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROSTATE - State
SPRMXCEF - Maximum Common Event Flag Wait

ELEMENT DERIVATION: SPRCEF is set to the value of the MNR_STA$L_CEF field from the VMS MONITOR STATES record.

RANGE OF VALUE: SPRCEF takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRCEF contains the last value of SPRCEF processed in the level of summarization.

USAGE NOTES: SPRCEF records how many processes are waiting for resources that are in global memory or are waiting on a system-wide event. If the number in SPRCEF represents a large percentage of the processes in the system, the system is probably locking too many global resources. These resources represent a bottleneck that may eventually degrade system performance.

| S P R C E N Q | Converted Enqueues

SPRCENQ contains the count of converted enqueues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSCEQ - Converted Enqueue Rate
SPRLCL - Lock Conversions, Local

ELEMENT DERIVATION: SPRCENQ is set to the value of the MNR_LCK$L_ENQCVT field of the VMS MONITOR LOCK record.

RANGE OF VALUE: SPRCENQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRCENQ contains the sum of all values of SPRCENQ processed in the level of summarization.

USAGE NOTES: SPRCENQ counts the number of requested locks (enqueues) that were granted after the process had been waiting.

| S P R C L S C T | CLS CLUSTER Record Count (Type 19)

SPRCLSCT contains a count of VMS MONITOR CLUSTER records processed.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS MONITOR CLUSTER record is stored, SPRCLSCT is set to 1.

RANGE OF VALUE: In the DETAIL timespan of the CA MICS VMS System Profile file, SPRCLSCT's value is always 0 or 1. At higher levels of summarization, its value ranges from 0 to the total number of VMS MONITOR CLUSTER records processed in the level of summarization.

SUMMARIZATION PROCESS: SPRCLSCT contains the sum of all values of SPRCLSCT processed in the level of summarization.

USAGE NOTES: SPRCLSCT is used as the divisor when computing averages and ratios with the measurement data.

| S P R C L S D R | CLS CLUSTER Record Duration (Type 19)

SPRCLSDR contains the number of seconds in the "reporting interval" of a given VMS MONITOR record. Reporting interval is the period of time during which a monitor gathers and reports statistics on this record.

TYPE: Common accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: SPRCLSCT - CLS CLUSTER Record Count (Type 19)
INTERVLS - Number of Recording Intervals
STARTTS - Start Time Stamp
ENDTS - End Time Stamp
DEXDUR - Requested Recording Interval Time

ELEMENT DERIVATION: SPRCLSDR contains the difference between the time stamp of the current VMS MONITOR record and the time stamp of the previously processed record of the same type.

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SPRCLSDR contains the sum of all occurrences of SPRCLSDR that were included in the level of summarization.

USAGE NOTES: Note that SPRCLSDR is uniquely calculated for this VMS MONITOR record type before merging records together into the DEMSPR file.

S P R C L S T M | CLS CPU Busy Time

SPRCLSTM contains the amount of time the CPU spent in all CPU modes excluding idle mode.

TYPE: Accumulated - Numeric, measurement unit is SAS time value in seconds.

SEE ALSO: SPRCPUTM - System CPU Busy Time

ELEMENT DERIVATION: SPRCLSTM is set to the value of the MNR_CLU$L_CPU_BUSY field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRCLSTM is a SAS time value ranging from 00.00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: SPRCLSTM contains the sum of all values of SPRCLSTM processed in the level of summarization.

USAGE NOTES: SPRCLSTM is the amount of time that the CPU was doing work. This time is an aggregate of all the CPU modes (executive, kernel, supervisor, and user) in which the processor operates. The difference between SPRCLSTM and DURATION (measurement interval) is the amount of time the processor was idle.

S P R C N T A B | Transactions Aborted Count

SPRCNTAB contains the count of transaction operations aborted by DECdtm services. It is the total of planned and unplanned aborts.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSTAB - Transaction Abort Rate
SPRMSTAB - Max Transaction Abort Rate

ELEMENT DERIVATION: SPRCNTAB is set to the value of the MNR_TRA$L_ABORTS field of the VMS MONITOR TRANSACTION record.

RANGE OF VALUE: SPRCNTAB ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRCNTAB contains the sum of all values of SPRCNTAB processed in the level of summarization.

USAGE NOTES: The Abort Transaction system service can be called by any resource manager or transaction manager participating in the transaction. This might be due to a machine failure, for example. In the abort phase, all transaction participants are ordered to abort and roll back their work.

S P R C N T A D | Add Remote Branch Ops Count

SPRCNTAD contains the count of add remote (to a remote subordinate parent) branch operations by DECdtm services.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSTAD - Transaction Remote Add Rate
SPRMSTAD - Max Transaction Remote Add Rate

ELEMENT DERIVATION: SPRCNTAD is set to the value of the MNR_TRA$L_ADDS field of the VMS MONITOR TRANSACTION record.

RANGE OF VALUE: SPRCNTAD ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRCNTAD contains the sum of all values of SPRCNTAD processed in the level of summarization.

USAGE NOTES: None

| S P R C N T B R | Start Remote Branch Ops Count

SPRCNTBR contains the count of start remote (to a remote parent) branch operations by DECdtm services.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSTBR - Transaction Remote Start Rate
SPRMSTBR - Max Transaction Remote Start Rate

ELEMENT DERIVATION: SPRCNTBR is set to the value of the MNR_TRA$L_BRANCHS field of the VMS MONITOR TRANSACTION record.

RANGE OF VALUE: SPRCNTBR ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRCNTBR contains the sum of all values of SPRCNTBR processed in the level of summarization.

USAGE NOTES: None

| S P R C N T C O | Transactions Total Committed Count

SPRCNTCO contains the count of transactions committed in total. It is the combined total of one-phase and two-phase commit operations completed by DECdtm services.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSTCO - Transaction Total Commit Rate
SPRMSTCO - Max Transaction Total Commit Rate

ELEMENT DERIVATION: SPRCNTCO is set to the value of the MNR_TRA$L_COMMITS field of the VMS MONITOR TRANSACTION record.

RANGE OF VALUE: SPRCNTCO ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRCNTCO contains the sum of all values of SPRCNTCO processed in the level of summarization.

USAGE NOTES: None

| S P R C N T D 1 | Transactions < 1 Second Count

SPRCNTD1 contains the count of transactions with a duration of less than 1 second.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SPRCNTD1 is set to the value of the MNR_TRA$L_BUCKETS1 field of the VMS MONITOR TRANSACTION record.

RANGE OF VALUE: SPRCNTD1 ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRCNTD1 contains the sum of all values of SPRCNTD1 processed in the level of summarization.

USAGE NOTES: None

| S P R C N T D 2 | Transactions 1-2 Seconds Count

SPRCNTD2 contains the count of transactions with a duration of 1 second to less than 2 seconds.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SPRCNTD2 is set to the value of the MNR_TRA$L_BUCKETS2 field of the VMS MONITOR TRANSACTION record.

RANGE OF VALUE: SPRCNTD2 ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRCNTD2 contains the sum of all values of SPRCNTD2 processed in the level of summarization.

USAGE NOTES: None

```
+-----------------+
| S P R C N T D 3 |  Transactions 2-3 Seconds Count
+-----------------+
SPRCNTD3 contains the count of transactions with a duration
of 2 seconds to less than 3 seconds.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  SPRCNTD3 is set to the value of the
MNR_TRA$L_BUCKETS3 field of the VMS MONITOR TRANSACTION
record.

RANGE OF VALUE:  SPRCNTD3 ranges from a minimum of 0 to a
maximum limited only by practical bounds.

SUMMARIZATION PROCESS:  SPRCNTD3 contains the sum of all
values of SPRCNTD3 processed in the level of summarization.

USAGE NOTES:  None


+-----------------+
| S P R C N T D 4 |  Transactions 3-4 Seconds Count
+-----------------+
SPRCNTD4 contains the count of transactions with a duration
of 3 seconds to less than 4 seconds.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  SPRCNTD4 is set to the value of the
MNR_TRA$L_BUCKETS4 field of the VMS MONITOR TRANSACTION
record.

RANGE OF VALUE:  SPRCNTD4 ranges from a minimum of 0 to a
maximum limited only by practical bounds.

SUMMARIZATION PROCESS:  SPRCNTD4 contains the sum of all
values of SPRCNTD4 processed in the level of summarization.

USAGE NOTES:  None


+-----------------+
| S P R C N T D 5 |  Transactions 4-5 Seconds Count
+-----------------+
SPRCNTD5 contains the count of transactions with a duration
of 4 seconds to less than 5 seconds.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  SPRCNTD5 is set to the value of the
MNR_TRA$L_BUCKETS5 field of the VMS MONITOR TRANSACTION
record.

RANGE OF VALUE:  SPRCNTD5 ranges from a minimum of 0 to a
maximum limited only by practical bounds.

SUMMARIZATION PROCESS:  SPRCNTD5 contains the sum of all
values of SPRCNTD5 processed in the level of summarization.

USAGE NOTES:  None


+-----------------+
| S P R C N T D 6 |  Transactions > 5 Seconds Count
+-----------------+
SPRCNTD6 contains the count of transactions with a duration
of 5 seconds or greater.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  SPRCNTD6 is set to the value of the
MNR_TRA$L_BUCKETS6 field of the VMS MONITOR TRANSACTION
record.

RANGE OF VALUE:  SPRCNTD6 ranges from a minimum of 0 to a
maximum limited only by practical bounds.

SUMMARIZATION PROCESS:  SPRCNTD6 contains the sum of all
values of SPRCNTD6 processed in the level of summarization.

USAGE NOTES:  None
```

```
+-----------------+
| S P R C N T E N |  Transactions Ended Count
+-----------------+
SPRCNTEN contains the count of transaction operations ended
by DECdtm services.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSTEN - Transaction End Rate
           SPRMSTEN - Max Transaction End Rate

ELEMENT DERIVATION:  SPRCNTEN is set to the value of the
MNR_TRA$L_ENDS field of the VMS MONITOR TRANSACTION record.

RANGE OF VALUE:  SPRCNTEN ranges from a minimum of 0 to a
maximum limited only by practical bounds.

SUMMARIZATION PROCESS:  SPRCNTEN contains the sum of all
values of SPRCNTEN processed in the level of summarization.

USAGE NOTES:  SPRCNTEN reflects the number of times that the
$END_TRANS system service completed.


+-----------------+
| S P R C N T O P |  One-Phase Commits Initiated
+-----------------+
SPRCNTOP contains the count of one-phase commit operations
initiated by DECdtm services.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSTOP - Transaction One-Phase Init Rate
           SPRMSTOP - Max Transaction One-Phase Init Rate

ELEMENT DERIVATION:  SPRCNTOP is set to the value of the
MNR_TRA$L_ONE_PHASE field of the VMS MONITOR TRANSACTION
record.

RANGE OF VALUE:  SPRCNTOP ranges from a minimum of 0 to a
maximum limited only by practical bounds.

SUMMARIZATION PROCESS:  SPRCNTOP contains the sum of all
values of SPRCNTOP processed in the level of summarization.

USAGE NOTES:  SPRCNTOP reflects the number of transactions
processed using a lower resource operation because there is
only one resource manager participating in the transaction.
```

```
+-----------------+
| S P R C N T P R |  Transactions Prepared Count
+-----------------+
SPRCNTPR contains the count of transactions placed in the
prepare state by DECdtm services.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSTPR - Transaction Prepare Rate
           SPRMSTPR - Max Transaction Prepare Rate

ELEMENT DERIVATION:  SPRCNTPR is set to the value of the
MNR_TRA$L_PREPARES field of the VMS MONITOR TRANSACTION
record.

RANGE OF VALUE:  SPRCNTPR ranges from a minimum of 0 to a
maximum limited only by practical bounds.

SUMMARIZATION PROCESS:  SPRCNTPR contains the sum of all
values of SPRCNTPR processed in the level of summarization.

USAGE NOTES:  The prepare phase is the first phase of the
two-phase commit protocol implemented by DECdtm services when
the $END_TRANS service is called.


+-----------------+
| S P R C N T S T |  Transactions Started Count
+-----------------+
SPRCNTST contains the count of transaction operations started
by DECdtm services.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSTST - Transaction Start Rate
           SPRMSTST - Max Transaction Start Rate

ELEMENT DERIVATION:  SPRCNTST is set to the value of the
MNR_TRA$L_STARTS field of the VMS MONITOR TRANSACTION record.

RANGE OF VALUE:  SPRCNTST ranges from a minimum of 0 to a
maximum limited only by practical bounds.

SUMMARIZATION PROCESS:  SPRCNTST contains the sum of all
values of SPRCNTST processed in the level of summarization.

USAGE NOTES:  SPRCNTST reflects the number of successful
```

completions of the $START_TRANS system service.

| S P R C O L P G | Collided Page Wait

SPRCOLPG contains the count of processes waiting for pages of memory that have been placed in the system page file.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPFW - Page Fault Wait
SPRFPGWT - Free page wait
SPRMXCPW - Maximum Collided Page Wait

ELEMENT DERIVATION: SPRCOLPG is set to the value of the MNR_STA$L_COLPG field of the VMS MONITOR STATES record.

RANGE OF VALUE: SPRCOLPG takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRCOLPG contains the last value of SPRCOLPG processed in the level of summarization.

USAGE NOTES: A high value in SPRCOLPG indicates that the process working set sizes are not set high enough. If there is still memory on the free page list, the process working sets should be increased to help lower collided paging. Collided paging occurs when a process requests access to a page of memory that had been in memory at one time but has been written to the page file to free memory to satisfy other memory accesses.

If SPRCOLPG is high and there is no memory on the free page list, the system may need additional physical memory. If SPRCOLPG is high, the free page list is nearly empty and only a few processes are outswapped, the system is probably tuned as well as it can be for the limited resources available.

A high value in SPRCOLPG will have negative effects on the CPU and the I/O subsystem. The CPU usage will not show in any of the processes on the system but will be recorded as kernel and executive mode processing.

| S P R C O M | Compute State Wait, Inswapped

SPRCOM contains the count of processes that are ready to compute but are waiting to be scheduled by VMS.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROSTATE - State
SPRMXCSI - Maximum Compute State Wait, Inswapped
SPRCOMO - Compute State Wait, Outswapped

ELEMENT DERIVATION: SPRCOM is set to the value of the MNR_STA$L_COM field from the VMS MONITOR STATES record.

RANGE OF VALUE: SPRCOM takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRCOM contains the last value of SPRCOM processed in the level of summarization.

USAGE NOTES: VMS allows a running system to support more active processes than can fit into physical memory at one time. It does this by moving entire processes in an out of physical memory. Inactive processes are moved from physical memory to swap files on disk (outswapped). When they become active, VMS moves them back into physical memory (inswapped). SPRCOM records how many processes are ready to compute but are prevented from doing so because the CPU is busy with higher priority tasks. If the number in SPRCOM is a significant percentage of the total number of processes on the system, the CPU is not fast enough for the workload.

| S P R C O M O | Compute State Wait, Outswapped

SPRCOMO contains the count of processes that are ready to compute, are waiting to be scheduled, and have been swapped out.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROSTATE - State
SPRMXCSO - Maximum Compute State Wait, Outswapped
SPRCOM - Compute State Wait, Inswapped

ELEMENT DERIVATION: SPRCOMO is set to the value of the MNR_STA$L_COMO field from the VMS MONITOR STATES record.

RANGE OF VALUE: SPRCOMO takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRCOMO contains the last value of SPRCOMO processed in the level of summarization.

USAGE NOTES: VMS allows a running system to support more active processes than can fit into physical memory at one time. It does this by moving entire processes in an out of physical memory. Inactive processes are moved from physical memory to swap files on disk (outswapped). When they become active, VMS moves them back into physical memory (inswapped). SPRCOMO records how many processes were ready to compute but were prevented from doing so because the scheduler swapped them out before the CPU was free to process them. If SPRCOMO is nonzero the system probably needs improvements. The most common reason for a nonzero value in SPRCOMO is that there is not enough physical memory. A second reason is that the CPU is not fast enough for the workload, but that may be due to excessive paging and swapping.

S P R C P U N I — Number of Instructions Executed

SPRCPUNI contains an estimate of the total number of CPU instructions executed, calculated by multiplying the measured CPU time by a user-supplied VUPS (where VAX Units of Power for VAX 11/780=1) rate.

TYPE: Accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SPRCPUTM*MIPS

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SPRCPUNI contains the sum of all occurrences of SPRCPUNI that were included in the level of summarization.

USAGE NOTES: SPRCPUNI is an approximation based on system engineering concepts that were valid for second-generation equipment with hard-wired instructions and single-threaded execution. The accuracy of this element is a function of the instruction mix used in the program, the contending load (if any) using the system concurrently with the step, the processor architecture, the system microcode level, and the MIPS rate supplied by the user during the CA MICS generation process. Altering any of these factors will change the reported value of SPRCPUNI for replicate runs, even though the exact code executes each time a step is initiated.

When used with caution, SPRCPUNI can be helpful in comparing the relative efficiency of different software realizations of the same function (e.g., the relative path lengths of two database management systems for a backup/reload cycle performed on equivalent collections of data).

S P R C P U T M — System CPU Busy Time

SPRCPUTM contains the amount of time the CPU spent in all CPU modes, excluding idle mode.

TYPE: Accumulated - Numeric, measurement unit is SAS time value in seconds.

SEE ALSO: MODINTTM - Interrupt Stack Time
MODMPSTM - Multi-Processor Sync Time
MODKERTM - Kernel Mode Time
MODEXETM - Executive Mode Time
MODSUPTM - Supervisor Mode Time
MODUSRTM - User Mode Time
MODCOMTM - Compatibility Mode Time

ELEMENT DERIVATION: SPRCPUTM is set to the value of the MNR_SYS$L_BUSY field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE: SPRCPUTM is a SAS time value ranging from 00.00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: SPRCPUTM contains the sum of all values of SPRCPUTM processed in the level of summarization.

USAGE NOTES: SPRCPUTM is the amount of time that the CPU was doing work. This time is the aggregate of all the CPU modes (executive, kernel, supervisor, and user) in which the processor operates. The difference between SPRCPUTM and DURATION (measurement interval) is the amount of time the processor was idle.

```
+-----------------+
| S P R C U R     |  Current Processes
+-----------------+
SPRCUR contains the number of current processes on the VMS
system.

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  PROSTATE - State
           SPRMXCP  - Maximum Current Processes

ELEMENT DERIVATION:  SPRCUR is set to the value of the
MNR_STA$L_CUR field of the VMS MONITOR STATES record.

RANGE OF VALUE:  SPRCUR takes any integer value from 1 to the
number of processors installed in the VAX hardware.

SUMMARIZATION PROCESS:  SPRCUR contains the last value of
SPRCUR processed in the level of summarization.

USAGE NOTES:  In a single processor VAX system, SPRCUR will
always be 1--the process running VMS MONITOR itself.  In a
multiprocessor system, SPRCUR can be as high as the number of
processors installed in the hardware.  If the value in SPRCUR
is less than the number of processors, then the workload is
not sufficient to keep the processors busy.


+-----------------+
| S P R C U R L K |  Current Locks
+-----------------+
SPRCURLK contains the number of locks currently on the
system during DURATION measurement interval.

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRCURRS - Current Resources
           SPRMXCLK - Maximum Current Locks

ELEMENT DERIVATION:  SPRCURLK is set to the value of the
MNR_LCK$L_NUMLOCK field of the VMS MONITOR LOCK record.

RANGE OF VALUE:  SPRCURLK takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRCURLK contains the last value of
SPRCURLK processed in the level of summarization.

USAGE NOTES:  SPRCURLK counts the number of resources on the
system that have been exclusively granted to processes.


+-----------------+
| S P R C U R R S |  Current Resources
+-----------------+
SPRCURRS contains the number of resources on which processes
can enqueue to gain their exclusive use.

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRCURLK - Current Locks
           SPRMXCRS - Maximum Current Resources

ELEMENT DERIVATION:  SPRCURRS is set to the value of the
MNR_LCK$L_NUMRES field of the VMS MONITOR LOCK record.

RANGE OF VALUE:  SPRCURRS takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRCURRS contains the last value of
SPRCURRS processed in the level of summarization.

USAGE NOTES:  SPRCURRS counts the current number of enqueue
control blocks registered with the Lock Manager.  When a
process wants exclusive access to a system-wide resource, it
requests an enqueue control block associated with the
resource.  The resource is then counted as both a current
resource and as a current lock (SPRCURLK).  When the process
is done with exclusive use of the resource, it dequeues the
resource, removing it from the list of current locks.  The
enqueue control block remains associated with the resource
until the control block is deregistered.


+-----------------+
| S P R D A T A T |  Directory Data Cache Attempts
+-----------------+
SPRDATAT contains the number of times the system attempted to
locate a file name in the directory data cache.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRDATHT - Directory Data Cache Hits

ELEMENT DERIVATION:  SPRDATAT is set to the value of the
MNR_FIL$L_DIRDATA_TRIES field of the VMS MONITOR
FILE_SYSTEM_CACHE record.
```

RANGE OF VALUE: SPRDATAT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDATAT contains the sum of all values of SPRDATAT processed in the level of summarization.

USAGE NOTES: When a file is opened, created, closed, deleted, extended, or truncated, the file name is looked up in the directory. The directory cache is searched for the directory data needed. The occurrence is recorded in SPRDATAT. If the directory is found in the cache, the occurrence is recorded in SPRDATHT. The directory data cache size is set by the SYSGEN parameter ACP_DIRCACHE.

S P R D A T H T | Directory Data Cache Hits

SPRDATHT contains the number of successful attempts to find data in the directory data cache.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRDATAT - Directory Data Cache Attempts

ELEMENT DERIVATION: SPRDATHT is set to the value of the MNR_FIL$L_DIRDATA_HIT field of the VMS MONITOR FILE_SYSTEM_CACHE record.

RANGE OF VALUE: SPRDATHT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDATHT contains the sum of all values of SPRDATHT processed in the level of summarization.

USAGE NOTES: When a file is opened, created, deleted, extended, or truncated, the file name is looked up in the directory. The directory cache is searched for the directory data needed. The occurrence of the search is recorded in SPRDATAT. If the directory is found in the cache, the occurrence is recorded in SPRDATHT. The directory data cache size is set by the SYSGEN parameter ACP_DIRCACHE.

S P R D E A D F | Deadlocks Found

SPRDEADF contains the number of times the Lock Manager found an enqueue deadlock.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRDEADS - Deadlock Searches
SPRPSDFD - Deadlocks Found Rate

ELEMENT DERIVATION: SPRDEADF is set to the value of the MNR_LCK$L_DLCKFND field of the VMS MONITOR LOCK record.

RANGE OF VALUE: SPRDEADF takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDEADF contains the sum of all values of SPRDEADF processed in the level of summarization.

USAGE NOTES: Deadlocks are a potential problem any time processes must serialize resources to accomplish their tasks. Whenever a process holds a resource beyond the time of the system deadlock wait, the Lock Manager searches all processes waiting on resources to determine if a deadlock condition exists. SPRDEADF is the number of times the Lock Manager determined that a deadlock existed and took measures to clear the blocked resource.

S P R D E A D S | Deadlock Searches

SPRDEADS contains the number of times the Lock Manager performed a deadlock search.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRDEADF - Deadlocks Found
SPRPSDSR - Deadlock Search Rate
SPRDLM - Deadlock Messages

ELEMENT DERIVATION: SPRDEADS is set to the value of the MNR_LCK$L_DLKSRCH field of the VMS MONITOR LOCK record.

RANGE OF VALUE: SPRDEADS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDEADS contains the sum of all

values of SPRDEADS processed in the level of summarization.

USAGE NOTES: Deadlocks are a potential problem any time processes must serialize resource usage to accomplish their tasks. Whenever a process holds a resource beyond the time of the system deadlock wait, the Lock Manager searches all processes waiting for exclusive use of resources to determine if a deadlock condition exists. SPRDEADS is the number of times the Lock Manager searched for deadlocks.

| S P R D E Q | Dequeues

SPRDEQ contains the number of dequeues issued to release or free up resources.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSDEQ - Dequeue Rate
SPRUNL - Unlocks, Local

ELEMENT DERIVATION: SPRDEQ is set to the value of the MNR_LCK$L_DEQ field of the VMS MONITOR LOCK record.

RANGE OF VALUE: SPRDEQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDEQ contains the sum of all values of SPRDEQ processed in the level of summarization.

USAGE NOTES: None

| S P R D I O S | Direct I/Os

SPRDIOS contains the count of I/O requests to nonbuffered devices such as disks or tape drives.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRODIRIO - Direct I/Os
SPRDIRIO - Direct I/Os
PRCDIRIO - Direct I/Os
IMGDIRIO - Direct I/Os

ELEMENT DERIVATION: SPRDIOS is set to the value of the MNR_SYS$L_DIRIO field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE: SPRDIOS takes an integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDIOS contains the sum of all values of SPRDIOS processed in the level of summarization.

USAGE NOTES: SPRDIOS reflects the number of direct I/O operations, not the size of those operations. Direct I/Os can have great effect on system performance, but the effect varies widely depending on average transaction size. If the number of direct I/Os is constant, but the average transaction size is increasing, the negative effect will be greater.

| S P R D I R I O | Direct I/Os

SPRDIRIO contains the count of I/O requests to nonbuffered devices, such as disks or tape drives.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRODIRIO - Direct I/Os
SPRDIOS - Direct I/Os
PRCDIRIO - Direct I/Os
IMGDIRIO - Direct I/Os

ELEMENT DERIVATION: SPRDIRIO is set to the value of the MNR_IO$L_DIRIO field of the VMS MONITOR IO record.

RANGE OF VALUE: SPRDIRIO takes an integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDIRIO contains the sum of all values of SPRDIRIO processed in the level of summarization.

USAGE NOTES: SPRDIRIO reflects the number of direct I/O operations, not the size of those operations. Direct I/Os can have great effect on system performance, but the effect varies widely depending on the average transaction size. System performance can be degraded even when the number of direct I/Os is constant and average transaction size is increasing.

```
+-----------------+
| S P R D L K C T |  DLK DLOCK Record Count (Type 14)
+-----------------+
SPRDLKCT contains a count of VMS MONITOR DLOCK processed by
the CA MICS VAX/VMS Analyzer.

TYPE:  Accumulated - Numeric, measurement unit is integer.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  When a VMS MONITOR DLOCK record is
processed, SPRDLKCT is set to 1.

RANGE OF VALUE:  In the DETAIL timespan of the CA MICS VMS
System Profile file, SPRDLKCT is always 0 or 1.  At higher
levels of summarization, its value ranges from 0 to the total
number of VMS MONITOR DLOCK records processed in the level of
summarization.

SUMMARIZATION PROCESS:  SPRDLKCT contains the sum of all
values of SPRDLKCT processed in the level of summarization.

USAGE NOTES:  SPRDLKCT is used as a divisor for calculating
averages and ratios in the level of summarization.


+-----------------+
| S P R D L K D R |  DLK DLOCK Record Duration (Type 14)
+-----------------+
SPRDLKDR contains the number of seconds in the "reporting
interval" of a given VMS MONITOR record.  Reporting interval
is the period of time during which a monitor gathers and
reports statistics on this record.

TYPE:  Common accumulated data element - Numeric, measurement
unit is integer.

SEE ALSO:  SPRDLKCT - DLK DLOCK Record Count (Type 14)
           INTERVLS - Number of Recording Intervals
           STARTTS  - Start Time Stamp
           ENDTS    - End Time Stamp
           DEXDUR   - Requested Recording Interval Time

ELEMENT DERIVATION:  SPRDLKDR contains the difference between
the time stamp of the current VMS MONITOR record and the time
stamp of the previously processed record of the same type.

RANGE OF VALUE:  The value of this element ranges from a
minimum of 0 to a maximum that is limited only by practical
bounds.

SUMMARIZATION PROCESS:  SPRDLKDR contains the sum of all
occurrences of SPRDLKDR that were included in the level of
summarization.

USAGE NOTES:  Note that SPRDLKDR is uniquely calculated for
this VMS MONITOR record type before merging records together
into the DEMSPR file.


+-----------------+
| S P R D L M     |  Deadlock Messages
+-----------------+
SPRDLM contains the number of deadlock detection messages
processed by the Lock Manager.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRDEADS - Deadlock Searches

ELEMENT DERIVATION:  SPRDLM is set to the value of the
MNR_DLO$L_DLCKMSG field of the VMS MONITOR DLOCK record.

RANGE OF VALUE:  SPRDLM takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRDLM contains the sum of all values
of SPRDLM processed in the level of summarization.

USAGE NOTES:  Deadlocks among cluster nodes could become a
serious problem, so the Lock Manager sends messages to the
controlling process whenever a converted lock has been held
for longer than the SYSGEN parameter allows.  If the process
is waiting for another lock, it is told to go back to waiting
for this resource, and the resource is freed for another job.


+-----------------+
| S P R D M Z F L |  Demand Zero Page Faults
+-----------------+
SPRDMZFL contains the number of pages allocated and then set
to zeros as a result of page faults for DURATION (measurement
interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSDMZ - Demand Zero Page Fault Rate

ELEMENT DERIVATION:  SPRDMZFL is set to the value of the
```

MNR_PAG$L_DZROFLTS field of the VMS MONITOR PAGE record.

RANGE OF VALUE: SPRDMZFL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDMZFL contains the sum of all values of SPRDMZFL processed in the level of summarization.

USAGE NOTES: For each count in SPRDMZFL, a process requested a page of memory to be allocated for that process and for the system to initialize the page to zeros. This activity usually is performed when an image first begins using memory that had been logically allocated at compile time. Therefore, SPRDMZFL is a good estimator of the effect of image activation on paging.

Some images also dynamically request zeroed pages from the system, but on typical applications dynamic demand-zero faults are only a small portion of SPRDMZFL.

S P R D N T C T — DNT DECNET Record Count (Type 8)

SPRDNTCT contains a count of VMS MONITOR DECNET records processed by the CA MICS VAX/VMS Analyzer.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS MONITOR DECNET record is processed, SPRDNTCT is set to 1.

RANGE OF VALUE: In the DETAIL timespan of the CA MICS VMS System Profile file, SPRDNTCT is always 0 or 1. At higher levels of summarization, its value ranges from 0 to the total number of records processed in the level of summarization.

SUMMARIZATION PROCESS: SPRDNTCT contains the sum of all values of SPRDNTCT processed in the level of summarization.

USAGE NOTES: SPRDNTCT is used as a divisor for calculating averages and ratios in the level of summarization.

S P R D N T D R — DNT DECNET Record Duration (Type 8)

SPRDNTDR contains the number of seconds in the "reporting interval" of a given VMS MONITOR record. Reporting interval is the period of time during which a monitor gathers and reports statistics on this record.

TYPE: Common accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: SPRDNTCT - DNT DECNET Record Count (Type 8)
INTERVLS - Number of Recording Intervals
STARTTS - Start Time Stamp
ENDTS - End Time Stamp
DEXDUR - Requested Recording Interval Time

ELEMENT DERIVATION: SPRDNTDR contains the difference between the time stamp of the current VMS MONITOR record and the time stamp of the previously processed record of the same type.

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SPRDNTDR contains the sum of all occurrences of SPRDNTDR that were included in the level of summarization.

USAGE NOTES: Note that SPRDNTDR is uniquely calculated for this VMS MONITOR record type before merging records together into the DEMSPR file.

S P R D P L P K — Departing Local Packets

SPRDPLPK contains the number of departing local packets.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSDPL - Departing Local Packet Rate

ELEMENT DERIVATION: SPRDPLPK is set to the value of the MNR_NET$L_DEPLOCPK field of the VMS MONITOR DECNET record.

RANGE OF VALUE: SPRDPLPK takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDPLPK contains the sum of all

values of SPRDPLPK processed in the level of summarization.

USAGE NOTES: Departing local packets are DECnet data packets and control messages that originate with this node. It is presumed that the number of departing transit packages will be the same as those that arrived. The ratio between SPRARTPK and SPRDPLPK gives a rough estimate of the effect of routing on outgoing network communications.

| S P R D Q I | Dequeue Incoming

SPRDQI contains the count of incoming dequeues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRDQO - Dequeue Outgoing
SPRDQL - Dequeue Local

ELEMENT DERIVATION: SPRDQI is set to the value of the MNR_CLU$L_DEQIN field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRDQI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDQI contains the sum of all values of SPRDQI processed in the level of summarization.

USAGE NOTES: SPRDQI counts the number of locks from remote nodes that were released (dequeued).

| S P R D Q L | Dequeue Local

SPRDQL contains the count of local dequeues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRDQO - Dequeue Outgoing
SPRDQI - Dequeue Incoming

ELEMENT DERIVATION: SPRDQL is set to the value of the MNR_CLU$L_DEQLOC field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRDQL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDQL contains the sum of all values of SPRDQL processed in the level of summarization.

USAGE NOTES: SPRDQL counts the number of locks from this node that were released (dequeued).

| S P R D Q O | Dequeue Outgoing

SPRDQO contains the count of outgoing dequeues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRDQI - Dequeue Incoming
SPRDQL - Dequeue Local

ELEMENT DERIVATION: SPRDQO is set to the value of the MNR_CLU$L_DEQOUT field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRDQO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRDQO contains the sum of all values of SPRDQO processed in the level of summarization.

USAGE NOTES: SPRDQO counts the number of locks from this node that were released (dequeued) from remote nodes.

| S P R E C I | Enqueue Conversions Incoming

SPRECI contains the count of converted incoming enqueues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRECO - Enqueue Conversions Outgoing
SPRECL - Enqueue Conversions Local

ELEMENT DERIVATION: SPRECI is set to the value of the MNR_CLU$L_ENQCVTIN field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRECI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRECI contains the sum of all values

of SPRECI processed in the level of summarization.

USAGE NOTES: SPRECI counts the number of requested locks (enqueues) from remote nodes that were actually granted.

| S P R E C L | Enqueue Conversions Local

SPRECL contains the count of converted local enqueues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRECO - Enqueue Conversions Outgoing
SPRECI - Enqueue Conversions Incoming

ELEMENT DERIVATION: SPRECL is set to the value of the MNR_CLU$L_ENQCVTLOC field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRECL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRECL contains the sum of all values of SPRECL processed in the level of summarization.

USAGE NOTES: SPRECL counts the number of requested locks (enqueues) from this node that were actually granted.

| S P R E C O | Enqueue Conversions Outgoing

SPRECO contains the count of converted outgoing enqueues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRECI - Enqueue Conversions Incoming
SPRECL - Enqueue Conversions Local

ELEMENT DERIVATION: SPRECO is set to the value of the MNR_CLU$L_ENQCVTOUT field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRECO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRECO contains the sum of all values of SPRECO processed in the level of summarization.

USAGE NOTES: SPRECO counts the number of requested locks (enqueues) sent to remote nodes that were actually granted.

| S P R E N Q N Q | Enqueues Not Queued

SPRENQNQ contains the number of enqueue requests not queued because 'wait' was not specified with the request.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSENQ - Enqueues Not Queued Rate
SPRENQWT - Enqueue Waits

ELEMENT DERIVATION: SPRENQNQ is set to the value of the MNR_LCK$L_ENQNOTQD field from the VMS MONITOR LOCK record.

RANGE OF VALUE: SPRENQNQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRENQNQ contains the sum of all values of SPRENQNQ processed in the level of summarization.

USAGE NOTES: When a process requests an enqueue of a resource, it can specify what to do if the resource is already locked; it can wait until the resource becomes available or it can try again later. If the request specified it does not want to wait, the enqueue request will not be queued. The occurrence is recorded in SPRENQNQ. If SPRENQNQ is consistently large, there may be at least one process using program logic to determine when the resource becomes available, rather than using the conventional 'wait' option when requesting the enqueue. This can quickly consume CPU time and should be discouraged.

| S P R E N Q W T | Enqueue Waits

SPRENQWT contains the number of times enqueue requests were queued to wait for a resource to become available.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSEWT - Enqueue Waits Rate
SPRENQNQ - Enqueues Not Queued

ELEMENT DERIVATION: SPRENQWT is set to the value of the MNR_LCK$L_ENQWAIT field of the VMS MONITOR LOCK record.

RANGE OF VALUE: SPRENQWT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRENQWT contains the sum of all values of SPRENQWT processed in the level of summarization.

USAGE NOTES: When a process requests an enqueue of a resource, it can specify what to do if the resource is already locked; it can wait until the resource becomes available or it can try again later. If the request specified to wait, the request will be placed on a queue to wait for the resource to become available. This occurrence is recorded in SPRENQWT. If SPRENQWT is consistently large, there is a great deal of contention for a limited resource(s). Investigate which resource(s) is causing the bottleneck and make appropriate changes to reduce the contention.

S P R E R A S E — Erase I/O Operations

SPRERASE contains the number of requests the file system received to erase data in files during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSEIO - Erase I/O Rate

ELEMENT DERIVATION: SPRERASE is set to the value of the MNR_FCP$L_ field of the VMS MONITOR FCP record.

RANGE OF VALUE: SPRERASE takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRERASE contains the sum of all values of SPRERASE processed in the level of summarization.

USAGE NOTES: If SPRERASE is high and system I/O performance is poor, high-water marking should be turned off on the volumes affected. By default, VMS disks are initialized with high-water marking enabled. The high-water marking feature is typically used for security purposes because it triggers the file system to write zeros to all allocated bytes when a file is deleted. Although good for security, it could noticeably affect I/O performance on disks where files are frequently created and deleted.

S P R E X T A T — Extent Cache Attempts

SPREXTAT contains the number of attempts the system made to allocate file space in extent cache.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPREXTHT - Extent Cache Hits

ELEMENT DERIVATION: SPREXTAT is set to the value of the MNR_FIL$L_EXT_TRIES field of the VMS MONITOR FILE_SYSTEM_CACHE record.

RANGE OF VALUE: SPREXTAT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPREXTAT contains the sum of all values of SPREXTAT processed in the level of summarization.

USAGE NOTES: When space is allocated for a file, the first place searched is extent cache. The occurrence of the search is recorded in SPREXTAT. If there is a large enough space to satisfy the allocation, the space is allocated in extent cache and the occurrence is recorded in SPREXTHT. When a file is deleted or truncated, an attempt is made to put the space back into extent cache. This occurrence is recorded in SPREXTAT. If extent cache can hold the space being returned, the occurrence is recorded in SPREXTHT. If SPREXTAT is much larger than SPREXTHT, either the disk is fragmented or extent cache is too small. The size of extent cache is set by the SYSGEN parameters ACP_EXTCACHE and ACP_EXTLIMIT.

S P R E X T H T — Extent Cache Hits

SPREXTHT contains the number of times the system was successful at allocating space in or returning space to extent cache.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPREXTAT - Extent Cache Attempts

ELEMENT DERIVATION: SPREXTHT is set to the value of the MNR_FIL$L_EXT_HIT field of the VMS MONITOR FILE_SYSTEM_CACHE record.

RANGE OF VALUE: SPREXTHT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPREXTHT contains the sum of all values of SPREXTHT processed in the level of summarization.

USAGE NOTES: When space is allocated for a file, the first place searched is extent cache. The occurrence of the search is recorded in SPREXTAT. If there is a large enough space to satisfy the allocation, the space is allocated in extent cache and the occurrence is recorded in SPREXTHT. When a file is deleted or truncated, an attempt is made to put the space back into extent cache. This occurrence is recorded in SPREXTAT. If extent cache can hold the space being returned, the success is recorded in SPREXTHT. If SPREXTAT is much larger than SPREXTHT, either the disk is fragmented or extent cache is too small. The size of extent cache is set by the SYSGEN parameters ACP_EXTCACHE and ACP_EXTLIMIT.

+----------+
| S P R F C B A T | Directory FCB Cache Attempts
+----------+
SPRFCBAT contains the number of attempts to find information in the directory file control block (FCB) cache.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFCBHT - Directory FCB Cache Hits

ELEMENT DERIVATION: SPRFCBAT is set to the value of the MNR_FIL$L_DIRFCB_TRIES field of the VMS MONITOR FILE_SYSTEM_CACHE record.

RANGE OF VALUE: SPRFCBAT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFCBAT contains the sum of all values of SPRFCBAT processed in the level of summarization.

USAGE NOTES: When a file is opened, created, deleted, extended, or truncated, the system has to find the directory containing the file name. The directory FCB cache is searched for the FCB describing the file. This occurrence is recorded in SPRFCBAT. If the FCB is found, the occurrence is recorded in SPRFCBHT. The size of the directory FCB cache is set by the SYSGEN parameters ACP_SYSACC and ACP_DINDXCACHE.

+----------+
| S P R F C B H T | Directory FCB Cache Hits
+----------+
SPRFCBHT contains the number of times the system successfully located file names in the directory file control block (FCB) cache.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFCBAT - Directory FCB Cache Attempts

ELEMENT DERIVATION: SPRFCBHT is set to the value of the MNR_FIL$L_DIRFCB_HIT field of the VMS MONITOR FILE_SYSTEM_CACHE record.

RANGE OF VALUE: SPRFCBHT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFCBHT contains the sum of all values of SPRFCBHT processed in the level of summarization.

USAGE NOTES: When a file is opened, created, deleted, extended, or truncated, the system has to find the directory containing the file name. The directory FCB cache is searched for the FCB describing the file. This occurrence is recorded in SPRFCBAT. If the FCB is found, the occurrence is recorded in SPRFCBHT. The size of the directory FCB cache is set by the SYSGEN parameters ACP_SYSACC and ACP_DINDXCACHE.

+----------+
| S P R F C P C T | FCP Record Count (Type 5)
+----------+
SPRFCPCT contains the number of FCP records entered.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS MONITOR FCP record is stored, SPRFCPCT is set to 1.

RANGE OF VALUE: In the DETAIL timespan of the CA MICS VMS System Profile file, SPRFCPCT is always 0 or 1. At higher levels of summarization, its value ranges from 0 to the total

number of VMS MONITOR FCP records processed in the level of summarization.

SUMMARIZATION PROCESS: SPRFCPCT contains the sum of all values of SPRFCPCT processed in the level of summarization.

USAGE NOTES: SPRFCPCT is used as the divisor for calculating averages and ratios in the level of summarization.

S P R F C P D R — FCP Record Duration (Type 5)

SPRFCPDR contains the number of seconds in the "reporting interval" of a given VMS MONITOR record. Reporting interval is the period of time during which a monitor gathers and reports statistics on this record.

TYPE: Common accumulated data element - Numeric, measurement unit is INTEGER.

SEE ALSO:
- SPRFCPCT - FCP Record Count (Type 5)
- INTERVLS - Number of Recording Intervals
- STARTTS - Start Time Stamp
- ENDTS - End Time Stamp
- DEXDUR - Requested Recording Interval Time

ELEMENT DERIVATION: SPRFCPDR contains the difference between the time stamp of the current VMS MONITOR record and the time stamp of the previously processed record of the same type.

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SPRFCPDR contains the sum of all occurrences of SPRFCPDR that were included in the level of summarization.

USAGE NOTES: Note that SPRFCPDR is uniquely calculated for this VMS MONITOR record type before merging records together into the DEMSPR file.

S P R F C P F L — FCP Page Faults

SPRFCPFL contains the number of pages of memory accessed by the file system processes but were not in physical memory during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO:
- SPRPSFFL - FCP Page Fault Rate
- SPRPGEFL - PGE Page Faults
- SPRIOVFL - IO Page Faults
- SPRPGFLT - Page Faults
- PRCPAGFT - Page Faults
- IMGPAGFT - Page Faults

ELEMENT DERIVATION: SPRFCPFL is set to the value of the MNR_FCP$L_FAULT field of the VMS MONITOR FCP record.

RANGE OF VALUE: SPRFCPFL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFCPFL contains the sum of all values of SPRFCPFL processed in the level of summarization.

USAGE NOTES: If SPRFCPFL is large, the file system processing is being held up by paging. Working set sizes should be set so that SPRFCPFL remains relatively small.

S P R F C P R I — FCP Disk Read I/Os

SPRFCPRI contains the number of read requests processed by the file system during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO:
- SPRPSFRI - FCP Disk Read I/O Rate
- DSKIOOPS - Disk I/O Operations

ELEMENT DERIVATION: SPRFCPRI is set to the value of the MNR_FCP$L_READ field of the VMS MONITOR FCP record.

RANGE OF VALUE: SPRFCPRI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFCPRI contains the sum of all values of SPRFCPRI processed in the level of summarization.

USAGE NOTES: If SPRFCPRI plus SPRFCPWI is close to the sum of the DSKIOOPS for all disks on the node, file system caching is not working right. To improve this, check for fragmentation of the file system and increase the size of the ancillary control process (ACP) cache buffers. To set the

ACP cache buffers, modify the SYSGEN parameters that begin with "ACP_".

S P R F C P T M | FCP CPU Time

SPRFCPTM contains the number of seconds of CPU time used by the file system during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is SAS time value.

SEE ALSO: MODCPUTM - Total CPU Time

ELEMENT DERIVATION: SPRFCPTM is set to the value of the MNR_FCP$L_FCPCPU field of the VMS MONITOR FCP record.

RANGE OF VALUE: SPRFCPTM is a SAS time value ranging from 00.00.00 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: SPRFCPTM contains the sum of all values of SPRFCPTM processed in the level of summarization.

USAGE NOTES: If SPRFCPTM is a significant percentage of MODCPUTM, then the file system is affecting system CPU response. High SPRFCPTM is usually an indication that the file system caches are not large enough and that the window turn rate is too high.

S P R F C P W I | FCP Disk Write I/Os

SPRFCPWI contains the number of write requests processed by the file system during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSFWI - FCP Disk Write I/O Rate
DSKIOOPS - Disk I/O Operations

ELEMENT DERIVATION: SPRFCPWI is set to the value of the MNR_FCP$L_ field of the VMS MONITOR FCP record.

RANGE OF VALUE: SPRFCPWI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFCPWI contains the sum of all values of SPRFCPWI processed in the level of summarization.

USAGE NOTES: If SPRFCPRI plus SPRFCPWI is close to the sum of the DSKIOOPS for all disks on the node, file system caching is not working right. To improve this, check for fragmentation of the file system and increase the size of the ancillary control process (ACP) cache buffers. To set the ACP cache buffers, modify the SYSGEN parameters that begin with "ACP_".

S P R F H D A T | File Header Cache Attempts

SPRFHDAT contains the number of attempts to find a file header in the file header cache.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFHDHT - File Header Cache Hits

ELEMENT DERIVATION: SPRFHDAT is set to the value of the MNR_FIL$L_FILHDR_TRIES field of the VMS MONITOR FILE_SYSTEM_CACHE record.

RANGE OF VALUE: SPRFHDAT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFHDAT contains the sum of all values of SPRFHDAT processed in the level of summarization.

USAGE NOTES: When an open, close, extend, delete, or truncate operation is requested on a file, the file header is located by searching the file header cache. The search is recorded in SPRFHDAT. If the header is found, the occurrence is recorded in SPRFHDHT. The size of the file header cache is set by the SYSGEN parameter ACP_HDRCACHE.

S P R F H D H T | File Header Cache Hits

SPRFHDHT contains the number of times the file header was successfully found in the file header cache.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFHDAT - File Header Cache Attempts

ELEMENT DERIVATION: SPRFHDHT is set to the value of the MNR_FIL$L_DIRDATA_HIT of the VMS MONITOR FILE_SYSTEM_CACHE record.

RANGE OF VALUE: SPRFHDHT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFHDHT contains the sum of all values of SPRFHDHT processed in the level of summarization.

USAGE NOTES: When an open, close, extend, delete or truncate operation is requested on a file, the file header is located by searching the file header cache. The search occurrence is recorded in SPRFHDAT. If the header is found, the occurrence is recorded in SPRFHDHT. The size of the file header cache is set by the SYSGEN parameter ACP_HDRCACHE.

S P R F I D A T | File ID Cache Attempts

SPRFIDAT contains the number of attempts to obtain file IDs from and return file IDs to the file ID cache.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFIDHT - File ID Cache Hits

ELEMENT DERIVATION: SPRFIDAT is set to the value of the MNR_FIL$L_FID_TRIES field of the VMS MONITOR FILE_SYSTEM_CACHE record.

RANGE OF VALUE: SPRFIDAT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFIDAT contains the sum of all values of SPRFIDAT processed in the level of summarization.

USAGE NOTES: File ID numbers are assigned when files are created and returned on deletion. When a file is created, the file ID cache is searched for the next file ID to use. The search occurrence is recorded in SPRFIDAT. If the file ID is found, the occurrence is recorded in SPRFIDHT. When a file is deleted, its file ID is returned to the file ID cache. The return occurrence also is recorded in SPRFIDAT. If the file ID can be returned to the cache, the occurrence is recorded in SPRFIDHT. If SPRFIDAT is much larger than SPRFIDHT, the performance of creating and deleting files is hampered. The size of the file ID cache is set by the SYSGEN parameter ACP_FIDCACHE.

S P R F I D H T | File ID Cache Hits

SPRFIDHT contains the number of times file IDs were successfully obtained from and returned to the file ID cache.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFIDAT - File ID Cache Attempts

ELEMENT DERIVATION: SPRFIDHT is set to the value of the MNR_FIL$L_FID_HIT field of the VMS MONITOR FILE_SYSTEM_CACHE record.

RANGE OF VALUE: SPRFIDHT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFIDHT contains the sum of all values of SPRFIDHT processed in the level of summarization.

USAGE NOTES: File ID numbers are issued when files are created and returned on deletion. When a file is created, the file ID cache is searched for the next file ID. The search occurrence is recorded in SPRFIDAT. If the file ID is there, the occurrence is recorded in SPRFIDHT. When a file is deleted, its file ID is returned to the file ID cache. The return occurrence is recorded in SPRFIDAT. If the file ID can be returned to cache, the occurrence is recorded in SPRFIDHT. If SPRFIDAT is much larger than SPRFIDHT, the performance of creating and deleting files is hampered. The size of the file ID cache is set by the SYSGEN parameter ACP_FIDCACHE.

S P R F I L O P | Files Opened

SPRFILOP contains the count of files opened for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSFOP - Files Opened Rate

SPROPEN - Files Opened

ELEMENT DERIVATION: SPRFILOP is set to the value of the MNR_IO$L_OPENS field of the VMS MONITOR IO record.

RANGE OF VALUE: SPRFILOP takes an integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFILOP contains the sum of all values of SPRFILOP processed in the level of summarization.

USAGE NOTES: Opening files can consume a lot of resources, both in CPU time and the number of disk I/Os required. If the directory of the file is cached in memory, then the resource requirement is less. When a file is opened, its location is looked up in the directory. If the directory is not in cache, then it must be read from disk into cache. Once the location of the file is found, the file header is read into the file header cache, the file's table of block locations and index buffers are allocated and read from disk.

If SPRFILOP is large, then a lot of the machine resources are being used to initialize the files. If this seems to be affecting performance, the VAX system manager can do one of two things:
- o Increase directory caches, which takes away user memory.
- o Modify applications so they do not open as many files.

S P R F N I — Directory Functions Incoming

SPRFNI contains the count of incoming directory functions.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFNO - Directory Functions Outgoing

ELEMENT DERIVATION: SPRFNI is set to the value of the MNR_DLO$L_DIROUT field of the VMS MONITOR DLOCK record.

RANGE OF VALUE: SPRFNI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFNI contains the sum of all values of SPRFNI processed in the level of summarization.

USAGE NOTES: Cluster-wide directory functions, such as file creation and deletion, require a lot of coordination on the part of the Lock Manager. SPRFNI keeps track of those that come from remote nodes.

S P R F N O — Directory Functions Outgoing

SPRFNO contains the count of outgoing directory functions.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFNI - Directory Functions Incoming

ELEMENT DERIVATION: SPRFNO is set to the value of the MNR_DLO$L_DIRIN field of the VMS MONITOR DLOCK record.

RANGE OF VALUE: SPRFNO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFNO contains the sum of all values of SPRFNO processed in the level of summarization.

USAGE NOTES: Cluster-wide directory functions, such as file creation and deletion, require a lot of coordination on the part of the Lock Manager. SPRFNO keeps track of those that are sent to remote nodes.

S P R F P G F L — Free List Page Faults

SPRFPGFL contains the number of pages allocated from the free page list to satisfy process requests.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSFPL - Free Page List Fault Rate
SPRFPGWT - Free Page Wait

ELEMENT DERIVATION: SPRFPGFL is set to the value of the MNR_PAG$L_FREFLTS field of the VMS MONITOR PAGE record.

RANGE OF VALUE: SPRFPGFL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRFPGFL contains the sum of all values of SPRFPGFL processed in the level of summarization.

USAGE NOTES: SPRFPGFL represents the number of pages needed

from the free page list. Free pages are requested during image activation and when a process needs to expand its working set size. SPRFPGFL is usually proportional to the amount of image activation on the system. When the free page list is depleted, the process requesting a free page is put into free page wait state until a free page is available.

Free pages are reclaimed at image termination or when working sets are trimmed. Typical values for SPRPSFPL range from 0 to 30, depending on image activation. A larger amount is not necessarily harmful, although high image activation may well be.

```
+-----------------+
| S P R F P G W T |  Free Page Wait
+-----------------+
```

SPRFPGWT contains the count of processes waiting for pages of memory from the free page list.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

```
SEE ALSO:  PROSTATE - State
           SPRPFW   - Page Fault Wait
           SPRCOLPG - Collided Page Wait
           SPRMXFPW - Maximum Free Page Wait
```

ELEMENT DERIVATION: SPRFPGWT is set to the value of the MNR_STA$L_FPG field of the VMS MONITOR STATES record.

RANGE OF VALUE: SPRFPGWT takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRFPGWT contains the last value of SPRFPGWT processed in the level of summarization.

USAGE NOTES: A nonzero value in SPRFPGWT means that some processes are waiting for new pages from the free page list but none are currently available. Free pages will be reclaimed from processes that have exceeded their working set quota first and then by swapping out low-priority processes.

SPRFPGWT usually will not get very large unless processes activate many images and memory is constrained. Increasing the free list cache size in the SYSGEN parameters should decrease the number of processes that must wait for free pages.

```
+-----------------+
| S P R F R E     |  Free Page List Size
+-----------------+
```

SPRFRE contains the count of pages on the free page list at the end of DURATION (measurement interval).

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

```
SEE ALSO:  SPRMNIFP - IO Minimum Free Page Count
           SPRFRPGC - PGE Free Page Count
           SPRFREEP - IO Free Page Count
           SPRFPGWT - Free Page Wait
```

ELEMENT DERIVATION: SPRFRE is set to the value of the MNR_CLU$L_FRLIST field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRFRE takes an integer value from 0 to the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS: SPRFRE contains the last value of SPRFRE processed in the level of summarization.

USAGE NOTES: SPRFRE represents the number of 512-byte pages of physical memory available for processes to allocate. The number is decremented whenever a free page list fault occurs (SPRFPGFL). SPRFRE is incremented when:

- o A process terminates
- o A process voluntarily frees memory
- o The memory manager decrements the working set size of a process
- o A large process is swapped out and the processes being swapped in are not as large as the hole left behind

The SYSGEN parameter FREELIM is used to signal a lower bound for SPRFRE. If FREELIM is ever larger than SPRFRE, the memory manager begins decrementing working set sizes to bring SPRFRE back up to FREEGOAL, another SYSGEN parameter. If SPRFRE is consistently less than FREEGOAL, there is probably not enough physical memory on the system.

If SPRFRE becomes 0, free page faults will also cause the faulting process to enter the free page wait state, SPRFPGWT. This always means that the system needs more physical memory.

```
+-----------------+
| S P R F R E E P |  IO Free Page Count
+-----------------+
SPRFREEP contains the count of pages on the free page list at
the end of DURATION (measurement interval).

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRMNIFP - IO Minimum Free Page Count
           SPRFRPGC - PGE Free Page Count
           SPRFRPG  - Free Page Count
           SPRFPGWT - Free Page Wait

ELEMENT DERIVATION:  SPRFREEP is set to the value of the
MNR_IO$L_FREECNT field of the VMS MONITOR IO record.

RANGE OF VALUE:  SPRFREEP takes an integer value from 0 to
the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS:  SPRFREEP contains the last value of
SPRFREEP processed in the level of summarization.

USAGE NOTES:  SPRFREEP represents the number of 512-byte
pages of physical memory available for processes to allocate.
The number is decremented whenever a free page list fault
(SPRFPGFL) occurs.  SPRFREEP is incremented when:
 o A process terminates
 o A process voluntarily frees memory
 o The memory manager decrements the working set size of a
   process
 o A large process is swapped out and the processes being
   swapped in are not as large as the hole left behind

The SYSGEN parameter FREELIM is used to signal a lower bound
for SPRFREEP.  If FREELIM is ever larger than SPRFREEP, the
memory manager begins decrementing working set sizes to bring
SPRFREEP back up to FREEGOAL, another SYSGEN parameter.  If
SPRFREEP is consistently less than FREEGOAL, there is
probably not enough physical memory on the system.

If SPRFREEP becomes 0, free page faults also will cause the
faulting process to enter the free page wait state, SPRFPGWT.
This always means that the system needs more physical memory.


+-----------------+
| S P R F R G M T |  MSCP Fragments
+-----------------+
SPRFRGMT contains the count of extra fragments read to honor
requests.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRMSCSP - MSCP Splits

ELEMENT DERIVATION:  SPRFRGMT is set to the value of the
MNR_MSC$L_FRAGMENT field of the VMS MONITOR MSCP_SERVER
record.

RANGE OF VALUE:  SPRFRGMT takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRFRGMT contains the sum of all
values of SPRFRGMT processed in the level of summarization.

USAGE NOTES:  When MSCP server requests are too large to be
served by the buffers, the server splits the requests into
smaller fragments.  For example, if the current request is
for 60 blocks but the buffers can only handle 16-block
requests, the current request will be split into 4 separate
fragments.  Three fragments will contain 16 blocks each and
one fragment will contain 14 blocks.  In this example,
SPRFRGMT would be incremented by 3, the number of extra
fragments needed.



+-----------------+
| S P R F R P G   |  Free Page Count
+-----------------+
SPRFRPG contains the count of pages on the free page list at
the end of DURATION (measurement interval).

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRMNIFP - IO Minimum Free Page Count
           SPRFRPGC - PGE Free Page Count
           SPRFREEP - IO Free Page Count
           SPRFPGWT - Free Page Wait

ELEMENT DERIVATION:  SPRFRPG is set to the value of the
MNR_SYS$L_FREECNT field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE:  SPRFRPG takes an integer value from 0 to the
number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS:  SPRFRPG contains the last value of
SPRFRPG processed in the level of summarization.

USAGE NOTES:  SPRFRPG represents the number of 512-byte pages
of physical memory available for processes to allocate.  The
```

number is decremented whenever a free page list fault (SPRFPGFL) occurs. SPRFRPG is incremented when:

- o A process terminates
- o A process voluntarily frees memory
- o The memory manager decrements the working set size of a process
- o A large process is swapped out and the processes being swapped in are not as large

The SYSGEN parameter FREELIM is used to signal a lower bound for SPRFRPG. If FREELIM is ever larger than SPRFRPG, the memory manager begins decrementing working set sizes to bring SPRFRPG back up to FREEGOAL, another SYSGEN parameter. If SPRFRPG is consistently less than FREEGOAL, there is probably not enough physical memory on the system.

If SPRFRPG becomes 0, free page faults will cause the faulting process to enter the free page wait state, SPRFPGWT. This always means that the system needs more physical memory.

```
+-----------------+
| S P R F R P G C |  PGE Free Page Count
+-----------------+
```

SPRFRPGC contains the count of pages on the free page list at the end of DURATION (measurement interval).

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMNPFP - PGE Minimum Free Page Count
SPRFPGFL - Free List Page Faults
SPRFREEP - IO Free Page Count
SPRFRPG - Free Page Count

ELEMENT DERIVATION: SPRFRPGC is set to the value of the MNR_PAG$L_FREECNT field of the VMS MONITOR PAGE record.

RANGE OF VALUE: SPRFRPGC takes any integer value from 0 to the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS: SPRFRPGC contains the last value of SPRFRPGC processed in the level of summarization.

USAGE NOTES: SPRFRPGC represents the number of 512-byte pages of physical memory available for processes to allocate. The number is decremented whenever a free page list fault (SPRFPGFL) occurs. SPRFRPGC is incremented when:

- o A process terminates
- o A process voluntarily frees memory
- o The memory manager decrements the working set size of a process
- o A large process is swapped out and the processes being swapped in are not as large as the hole left behind

The SYSGEN parameter FREELIM is used to signal a lower bound for SPRFRPGC. If FREELIM is ever larger than SPRFRPGC, the memory manager begins decrementing working set sizes to bring SPRFRPGC back up to FREEGOAL, another SYSGEN parameter. If SPRFRPGC is consistently less than FREEGOAL, there is probably not enough physical memory on the system.

If SPRFRPGC becomes 0, free page faults will also cause the faulting process to enter the free page wait state (SPRFPGWT). This always means that the system needs more physical memory.

```
+-----------------+
| S P R F S C C T |  FSC CACHE Record Count (Type 11)
+-----------------+
```

SPRFSCCT counts VMS MONITOR FILE_SYSTEM_CACHE records processed by the CA MICS VAX/VMS Analyzer.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS MONITOR FILE_SYSTEM_CACHE record is processed, SPRFSCCT is set to 1.

RANGE OF VALUE: In the DETAIL timespan of the CA MICS VMS System Profile file, SPRFSCCT is always 0 or 1. At higher levels of summarization, its value ranges from 0 to the total number of VMS MONITOR FSC records processed in the level of summarization.

SUMMARIZATION PROCESS: SPRFSCCT contains the sum of all values of SPRFSCCT processed in the level of summarization.

USAGE NOTES: SPRFSCCT is used as the divisor for calculating averages and ratios in the level of summarization.

| S P R F S C D R | FSC CACHE Record Duration (Type 11)

SPRFSCDR contains the number of seconds in the "reporting interval" of a given VMS MONITOR record. Reporting interval is the period of time during which a monitor gathers and reports statistics on this record.

TYPE: Common accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: SPRFSCCT - FSC CACHE Record Count (Type 11)
INTERVLS - Number of Recording Intervals
STARTTS - Start Time Stamp
ENDTS - End Time Stamp
DEXDUR - Requested Recording Interval Time

ELEMENT DERIVATION: SPRFSCDR contains the difference between the time stamp of the current VMS MONITOR record and the time stamp of the previously processed record of the same type.

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SPRFSCDR contains the sum of all occurrences of SPRFSCDR that were included in the level of summarization.

USAGE NOTES: Note that SPRFSCDR is uniquely calculated for this monitor record type before merging records together into the DEMSPR file.

| S P R G B L F L | Global Valid Page Faults

SPRGBLFL contains the number of page faults that were served by mapping shared memory that was already valid.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSGBL - Global Valid Page Fault Rate

ELEMENT DERIVATION: SPRGBLFL is set to the value of the MNR_PAG$L_GVALID field of the VMS MONITOR PAGE record.

RANGE OF VALUE: SPRGBLFL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRGBLFL contains the sum of all values of SPRGBLFL processed in the level of summarization.

USAGE NOTES: Global valid faults are the least expensive of the page faulting types. A high level in SPRGBLFL usually indicates a correspondingly high level of activation of images that have shared code or have a good deal of interprocess communications. It also can indicate a large amount of swapping of active processes, which is the least efficient form of swapping. So, even though global valid faults do not represent much of a system impediment, SPRGBLFL can signal that other problems may be negatively affecting system performance.

| S P R H I B | Hibernate Wait, Inswapped

SPRHIB contains the count of processes that have signaled the operating system that they will not process for some period of time.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROSTATE - State
SPRMXHWI - Maximum Hibernate Wait, Inswapped
SPRHIBO - Hibernate Wait, Outswapped

ELEMENT DERIVATION: SPRHIB is set to the value of the MNR_STA$L_HIB field from the VMS MONITOR STATES record.

RANGE OF VALUE: SPRHIB takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRHIB contains the last value of SPRHIB processed in the level of summarization.

USAGE NOTES: The count in SPRHIB does not affect VMS performance to any great extent. If the count is large and the maximum number of processes allowed on the system is reached, then processes that are hibernating take up process slots without doing any work at that time. Hibernating processes usually will do useful work when they are awakened by the system, so it is still important to have hibernating processes remain on the system.

```
+-----------------+
| S P R H I B O   |  Hibernate Wait, Outswapped
+-----------------+
SPRHIBO contains the count of processes that have signaled
the operating system that they will not process for some
period of time and the scheduler has caused them to be
swapped out.

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  PROSTATE - State
           SPRMXHWO - Maximum Hibernate Wait, Outswapped
           SPRHIB   - Hibernate Wait, Inswapped

ELEMENT DERIVATION:  SPRHIBO is set to the value of the
MNR_STA$L_HIBO field from the VMS MONITOR STATES record.

RANGE OF VALUE:  SPRHIBO takes any integer value from 0 to
the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS:  SPRHIBO contains the last value of
SPRHIBO processed in the level of summarization.

USAGE NOTES:  The count in SPRHIBO does not affect VMS
performance to any great extent.  If the count is large and
the maximum number of processes allowed on the system is
reached, then processes that are hibernating take up process
slots without doing any work at that time.  Hibernating
processes usually will do useful work when they are awakened
by the system, so it is still important to have hibernating
processes remain on the system.

If there are any outswapped processes on the system, it is
best that they be in the hibernate or the suspended state.
In those states, swapping has the least effect on
performance.


+-----------------+
| S P R H O L E S |  Nonpaged Pool Unused KBytes
+-----------------+
SPRHOLES contains the total amount (in kilobytes) of unused
space in the dynamic nonpaged pool.

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRMNHOL - Minimum Unused KBytes

ELEMENT DERIVATION:  SPRHOLES is set to the value of the
MNR_POO$L_HOLESUM field of the VMS MONITOR POOL record.

RANGE OF VALUE:  SPRHOLES takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRHOLES contains the last value of
SPRHOLES processed in the level of summarization.

USAGE NOTES:  As the nonpaged pool is used, areas that were
used and then freed are called holes.  The sum of the sizes
of the holes, measured in kilobytes, is stored in SPRHOLES.
When the nonpaged pool is depleted, memory allocations must
come from paged memory, which could negatively affect
performance for processes that use the resource extensively.


+-----------------+
| S P R I N S W P |  Inswaps
+-----------------+
SPRINSWP contains the count of working sets read into memory
from the swap file in DURATION (measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSINS - Inswap Rate

ELEMENT DERIVATION:  SPRINSWP is set to the value of the
MNR_IO$L_ISWPCNT field of the VMS MONITOR IO record.

RANGE OF VALUE:  SPRINSWP takes an integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRINSWP contains the sum of all
values of SPRINSWP processed in the level of summarization.

USAGE NOTES:  When there are a large number of working sets
being swapped in the system, performance will degrade.  If
SPRINSWP is large, the system may be thrashing--swapping out
a process to swap in another, then needing to swap the first
process back in immediately.  Thrashing usually occurs when
the system cannot make any good determination for swapping.
Thrashing is a sure signal that the system memory is too
small.  If SPRINSWP is large and SPRCOMO is not zero, the
system is thrashing on the swap file.

Swapping is highly I/O intensive because working sets are
often very large.  Additionally, it can increase CPU usage
because of swapper processing, which involves recalculating
page offsets, setting up asynchronous system traps for
interrupts, and the like.
```

```
+-----------------+
| S P R I O V C T |  IOV IO Record Count (Type 4)
+-----------------+
SPRIOVCT contains the count of VMS MONITOR IO records
processed by the CA MICS VAX/VMS Analyzer.

TYPE:  Accumulated - Numeric, measurement unit is integer.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  When a VMS MONITOR IO record is
processed, SPRIOVCT is set to 1.

RANGE OF VALUE:  In the DETAIL timespan of the CA MICS VMS
System Profile file, SPRIOVCT is always 1.  At higher levels
of summarization, its value ranges from 1 to the total number
of VMS MONITOR IO records processed in the level of
summarization.

SUMMARIZATION PROCESS:  SPRIOVCT contains the sum of all
values of SPRIOVCT processed in the level of summarization.

USAGE NOTES:  SPRIOVCT is used as the divisor for calculating
averages and ratios in the level of summarization


+-----------------+
| S P R I O V D R |  IOV IO Record Duration (Type 4)
+-----------------+
SPRIOVDR contains the number of seconds in the "reporting
interval" of a given VMS MONITOR record.  Reporting interval
is the period of time during which a monitor gathers and
reports statistics on this record.

TYPE:  Common accumulated data element - Numeric, measurement
unit is integer.

SEE ALSO:  SPRIOVCT - IOV IO Record Count (Type 4)
           INTERVLS - Number of Recording Intervals
           STARTTS  - Start Time Stamp
           ENDTS    - End Time Stamp
           DEXDUR   - Requested Recording Interval Time

ELEMENT DERIVATION:  SPRIOVDR contains the difference between
the time stamp of the current VMS MONITOR record and the time
stamp of the previously processed record of the same type.

RANGE OF VALUE:  The value of this element ranges from a
minimum of 0 to a maximum that is limited only by practical
bounds.

SUMMARIZATION PROCESS:  SPRIOVDR contains the sum of all
occurrences of SPRIOVDR that were included in the level of
summarization.

USAGE NOTES:  Note that SPRIOVDR is uniquely calculated for
this VMS MONITOR record type before merging records together
into the DEMSPR file.


+-----------------+
| S P R I O V F L |  IO Page Faults
+-----------------+
SPRIOVFL contains the count of pages allocated to processes
as a result of page faults for DURATION (measurement
interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSIFL - IO Page Fault Rate
           SPRPGEFL - PGE Page Faults
           SPRFCPFL - FCP Page Faults
           SPRPGFLT - Page Faults
           PRCPAGFT - Page Faults
           IMGPAGFT - Page Faults

ELEMENT DERIVATION:  SPRIOVFL is set to the value of the
MNR_IO$L_FAULTS field of the VMS MONITOR IO record.

RANGE OF VALUE:  SPRIOVFL takes an integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRIOVFL contains the sum of all
values of SPRIOVFL processed in the level of summarization.

USAGE NOTES:  High page faulting can cause negative effects
on the CPU and I/O subsystem of VAX/VMS.  To reduce page
faulting, the VAX system manager should accordingly adjust
the user authorization file parameters WSDEFAULT, WSQUOTA,
and WSEXTENT along with the SYSGEN parameters PFRATH,
BORROWLIM, and WSINC.

WSDEFAULT, WSQUOTA, and WSEXTENT describe the number of
physical pages allocated to a process.  WSDEFAULT is the
amount with which the process starts.  WSQUOTA is the amount
of memory allowed for the process, regardless of the memory
usage on the system.  WSEXTENT is the limit to the amount of
memory the process may borrow when there is memory to spare.
```

PFRATH is the number of page faults per 10-second interval that a process must exceed to be eligible to borrow free memory. BORROWLIM is the amount of free memory that must be available before any will be loaned. WSINC is the number of memory pages that are loaned with each increase.

S P R I P F R E | Intermediate Request Packets Free

SPRIPFRE contains the number of intermediate request packets (IRPs) free.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMNIPF - Minimum Intermediate Req Packets Free
SPRIPUSE - Intermediate Request Packets in Use

ELEMENT DERIVATION: SPRIPFRE is set to the value of the MNR_POO$L_IRPCNT field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRIPFRE takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRIPFRE contains the last value of SPRIPFRE processed in the level of summarization.

USAGE NOTES: The SYSGEN parameter IRPCOUNT determines how many (IRPs) are preallocated for use by processes on the system. When a process requests pool space that fits into an IRP, one is allocated to the process. SPRIPFRE is the count of IRPs that are available for use.

S P R I P U S E | Intermediate Request Packets in Use

SPRIPUSE contains the number of intermediate request packets (IRPs) in use.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMXIPU - Maximum Intermediate Req Packet in Use
SPRIPFRE - Intermediate Request Packets Free

ELEMENT DERIVATION: SPRIPUSE is set to the value of the MNR_POO$L_IRPINUSE field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRIPUSE takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRIPUSE contains the last value of SPRIPUSE processed in the level of summarization.

USAGE NOTES: The SYSGEN parameter IRPCOUNT determines how many (IRPs) are preallocated for use by processes on the system. When a process requests pool space that fits into an IRP, one is allocated to the process. SPRIPUSE is the count of IRPs that are in use.

S P R K B U S E | Nonpaged Pool KBytes in Use

SPRKBUSE contains the total amount (in kilobytes) of space used in the dynamic nonpaged pool.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMXKBU - Maximum Nonpaged Pool KBytes in Use

ELEMENT DERIVATION: SPRKBUSE is set to the value of the MNR_POO$L_DYNINUSE field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRKBUSE takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRKBUSE contains the last value of SPRKBUSE processed in the level of summarization.

USAGE NOTES: The dynamic portion of the nonpaged pool serves requests that do not closely fit the preallocated packets. It is also used as an overflow when all packets are in use. SPRKBUSE is the amount, in kilobytes, of space used from the dynamic nonpaged pool.

S P R L C I | Lock Conversions, Incoming

SPRLCI contains the count of converted incoming enqueues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRLCO - Lock Conversions, Outgoing

SPRLCL - Lock Conversions, Local

ELEMENT DERIVATION: SPRLCI is set to the value of the MNR_DLO$L_ENQCVTIN field of the VMS MONITOR DLOCK record.

RANGE OF VALUE: SPRLCI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRLCI contains the sum of all values of SPRLCI processed in the level of summarization.

USAGE NOTES: SPRLCI counts the number of requested locks (enqueues) from remote nodes that were actually granted.

| S P R L C L | Lock Conversions, Local

SPRLCL contains the count of converted local enqueues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRLCO - Lock Conversions, Outgoing
SPRLCI - Lock Conversions, Incoming

ELEMENT DERIVATION: SPRLCL is set to the value of the MNR_DLO$L_ENQCVTLOC field of the VMS MONITOR DLOCK record.

RANGE OF VALUE: SPRLCL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRLCL contains the sum of all values of SPRLCL processed in the level of summarization.

USAGE NOTES: SPRLCL counts the number of requested locks (enqueues) from this node that were actually granted.

| S P R L C O | Lock Conversions, Outgoing

SPRLCO contains the count of converted outgoing enqueues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRLCI - Lock Conversions, Incoming
SPRLCL - Lock Conversions, Local

ELEMENT DERIVATION: SPRLCO is set to the value of the MNR_DLO$L_ENQCVTOUT field of the VMS MONITOR DLOCK record.

RANGE OF VALUE: SPRLCO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRLCO contains the sum of all values of SPRLCO processed in the level of summarization.

USAGE NOTES: SPRLCO counts the number of requested locks (enqueues) sent to remote nodes that were actually granted.

| S P R L E F | Local Event Flag Wait, Inswapped

SPRLEF contains the count of processes waiting for a local event flag to be signaled.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROSTATE - State
SPRLEFO - Local Event Flag Wait, Outswapped
SPRMXLEI - Maximum Local Event Flag Wait, Inswap

ELEMENT DERIVATION: SPRLEF is set to the value of the MNR_STA$L_LEF field from the VMS MONITOR STATES record.

RANGE OF VALUE: SPRLEF takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRLEF contains the last value of SPRLEF processed in the level of summarization.

USAGE NOTES: SPRLEF records how many processes are waiting for resources that are locally created and managed. Most often, the processes in local event flag wait are waiting for user input at the terminal. Processes included in SPRLEF do not affect system performance much.

| S P R L E F O | Local Event Flag Wait, Outswapped

SPRLEFO contains the count of processes waiting for a local event flag to be signaled.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

```
SEE ALSO:  PROSTATE - State
           SPRLEF   - Local Event Flag Wait, Inswapped
           SPRMXLEO - Maximum Local Event Flag Wait, Outswap
                      outswapped

ELEMENT DERIVATION:  SPRLEFO is set to the value of the
MNR_STA$L_LEFO field from the VMS MONITOR STATES record.

RANGE OF VALUE:  SPRLEFO takes any integer value from 0 to
the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS:  SPRLEFO contains the last value of
SPRLEFO processed in the level of summarization.

USAGE NOTES:  SPRLEFO records how many processes are waiting
for resources that are locally created and managed.  Most
often, the processes in local event flag wait are waiting for
user input at the terminal.  Processes included in SPRLEFO do
not affect system performance much.

For a process included in SPRLEFO, when the local event
occurs, the performance of the process will be affected by
the need to swap the process back into memory.  This delay,
which directly affects the response the user sees, is the
only reason why a high count in SPRLEFO would require
attention.  If the users are not complaining about poor
response time, this value is of little importance.
```

S P R L O G N M — Logical Name Translations

SPRLOGNM contains the count of translations of logical names into physical file or device names.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSLNT - Logical Name Translation Rate

ELEMENT DERIVATION: SPRLOGNM is set to the value of the MNR_IO$L_LOGNAM field of the VMS MONITOR IO record.

RANGE OF VALUE: SPRLOGNM takes an integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRLOGNM contains the sum of all values of SPRLOGNM processed in the level of summarization.

USAGE NOTES: Logical names are translated to parse file names or access other system resources such as disk, tape, or network devices. Translation of logical name happens in kernel mode and is not charged to the process requesting translation. SPRLOGNM is usually proportional to image activation because the opening of files and devices for the new image involves logical name translation. If SPRLOGNM is high and MODKERTM is 25 percent or more of MODCPUTM, the applications probably should be restructured to not activate as many images. One key to reducing image activation is to reduce the use of DCL command files on the system.

S P R L O K — Total All Locks

SPRLOK contains the total number of lock operations on cluster resources.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

```
SEE ALSO:  SPRDQI - Dequeue Incoming
           SPRDQL - Dequeue Local
           SPRDQO - Dequeue Outgoing
           SPRECI - Enqueue Conversions Incoming
           SPRECL - Enqueue Conversions Local
           SPRECO - Enqueue Conversions Outgoing
           SPRNEI - New Enqueue Incoming
           SPRNEL - New Enqueue Local
           SPRNEO - New Enqueue Outgoing
```

ELEMENT DERIVATION: SPRLOK is set to the value of the MNR_CLU$L_TOTAL_LOCKS field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRLOK takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRLOK contains the sum of all values of SPRLOK processed in the level of summarization.

USAGE NOTES: SPRLOK is the sum of SPRNEI, SPRNEL, SPRNEO, SPRECI, SPRECL, SPRECO, SPRDQI, SPRDQL, and SPRDQO.

```
+-----------------+
| S P R L O K C T |  LOK LOCK Record Count (Type 7)
+-----------------+
SPRLOKCT contains a count of VMS MONITOR LOCK records
processed by the CA MICS VAX/VMS Analyzer.

TYPE:  Accumulated - Numeric, measurement unit is integer.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  When a VMS MONITOR LOCK record is
processed, SPRLOKCT is set to 1.

RANGE OF VALUE:  In the DETAIL timespan of the CA MICS VMS
System Profile file, SPRLOKCT is always 0 or 1.  At higher
levels of summarization, its value ranges from 0 to the total
number of VMS MONITOR LOCK records processed in the level of
summarization.

SUMMARIZATION PROCESS:  SPRLOKCT contains the sum of all
values of SPRLOKCT processed in the level of summarization.

USAGE NOTES:  SPRLOKCT is used as the divisor for calculating
averages and ratios in the level of summarization.


+-----------------+
| S P R L O K D R |  LOK LOCK Record Duration (Type 7)
+-----------------+
SPRLOKDR contains the number of seconds in the "reporting
interval" of a given VMS MONITOR record.  Reporting interval
is the period of time during which a monitor gathers and
reports statistics on this record.

TYPE:  Common accumulated data element - Numeric, measurement
unit is integer.

SEE ALSO:  SPRLOKCT - LOK LOCK Record Count (Type 7)
           INTERVLS - Number of Recording Intervals
           STARTTS  - Start Time Stamp
           ENDTS    - End Time Stamp
           DEXDUR   - Requested Recording Interval Time

ELEMENT DERIVATION:  SPRLOKDR contains the difference between
the time stamp of the current VMS MONITOR record and the time
stamp of the previously processed record of the same type.

RANGE OF VALUE:  The value of this element ranges from a
minimum of 0 to a maximum that is limited only by practical
bounds.

SUMMARIZATION PROCESS:  SPRLOKDR contains the sum of all
occurrences of SPRLOKDR that were included in the level of
summarization.

USAGE NOTES:  Note that SPRLOKDR is uniquely calculated for
this VMS MONITOR record type before merging records together
into the DEMSPR file.


+-----------------+
| S P R L P F R E |  Large Request Packets Free
+-----------------+
SPRLPFRE contains the number of large request packets (free.

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRMNLPF - Minimum Large Request Packets Free
           SPRLPUSE - Large Request Packets in Use

ELEMENT DERIVATION:  SPRLPFRE is set to the value of the
MNR_POO$L_LRPCNT field of the VMS MONITOR POOL record.

RANGE OF VALUE:  SPRLPFRE takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRLPFRE contains the last value of
SPRLPFRE processed in the level of summarization.

USAGE NOTES:  The SYSGEN parameter LRPCOUNT determines how
many large request packages are preallocated for use by
processes on the system.  When a process requests pool space
that fits into an LRP, one is allocated to the process.
SPRLPFRE is the count of LRPs that are available for use.


+-----------------+
| S P R L P U S E |  Large Request Packets in Use
+-----------------+
SPRLPUSE contains the number of large request packets (LRPs)
in use.

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRMXLPU - Maximum Large Request Packets in Use
           SPRLPFRE - Large Request Packets Free

ELEMENT DERIVATION:  SPRLPUSE is set to the value of the
MNR_POO$L_LRPINUSE field of the VMS MONITOR POOL record.
```

RANGE OF VALUE: SPRLPUSE takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRLPUSE contains the last value of SPRLPUSE processed in the level of summarization.

USAGE NOTES: The SYSGEN parameter LRPCOUNT determines how many (LRPs) are preallocated for use by processes on the system. When a process requests pool space that fits into an LRP, one is allocated to the process. SPRLPUSE is the count of LRPs that are in use.

S P R L R G P K Large Request Packets Left

SPRLRGPK contains the number of large request packets (LRPs) left.

This element will have missing values for systems running VMS Version 6.0 or later.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SPRLRGPK is set to the value of the MNR_NET$L_LRPCNT field of the VMS MONITOR DECNET record.

RANGE OF VALUE: SPRLRGPK takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRLRGPK contains the last value of SPRLRGPK processed in the level of summarization.

USAGE NOTES: LRPs are preallocated portions of the nonpaged pool. LRPs are primarily used in DECnet communications to store incoming DECnet packets. When LRPs are depleted, VMS adds more LRPs to the pool. The memory used for the new LRPs is taken from user memory. Messages received while new LRPs are being added will be retransmitted.

S P R M A I L W Mailbox Writes

SPRMAILW contains the count of writes to mailboxes.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSMBX - Mailbox Write Rate

ELEMENT DERIVATION: SPRMAILW is set to the value of the MNR_IO$L_MBWRITES field of the VMS MONITOR IO record.

RANGE OF VALUE: SPRMAILW takes an integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMAILW contains the sum of all values of SPRMAILW processed in the level of summarization.

USAGE NOTES: Mailboxes are mechanisms used for interprocess communication on VAX/VMS. When a process wants to receive messages from other processes, it sets up the mechanism to receive messages and then registers the mailbox with VMS, that places it on a list of mailboxes available. Processes wishing to send messages to that mailbox tell VMS where the message is and the mailbox to which it should be delivered. If the mailbox has room, VMS delivers the message and sends a low-priority interrupt to the receiving process. If the mailbox is full, VMS puts the message on a very high-priority interrupt queue that continually checks to see if the mailbox has room. If there are many undeliverable messages on the queue, these high-priority interrupts can block other, perhaps more important, interrupts from being processed in a timely manner. These interrupts are processed in kernel mode.

When SPRMAILW is high there is a higher likelihood of the mailbox full condition. If SPRMAILW is high and MODKERTM is 25 percent or more of MODCPUTM, the amount of mailbox writing is potentially harmful.

S P R M A S K Record Construction Audit Mask

SPRMASK is 13 bytes in length, with each byte signifying the presence or absence of a particular record type, as shown below:

BYTE - VALUE - RECORD TYPE INDICATED

```
     1 - 'S' or '.' - STA STATES Record (Type 1)
     2 - 'P' or '.' - PGE PAGE Record (Type 3)
     3 - 'I' or '.' - IOV IO Record (Type 4)
     4 - 'F' or '.' - FCP Record (Type 5)
     5 - 'P' or '.' - POL POOL Record (Type 6)
     6 - 'L' or '.' - LOK LOCK Record (Type 7)
     7 - 'D' or '.' - DNT DECNET Record (Type 8)
     8 - 'F' or '.' - FSC CACHE Record (Type 11)
     9 - 'D' or '.' - DLK DLOCK Record (Type 14)
    10 - 'S' or '.' - SYT SYSTEM Record (Type 17)
    11 - 'C' or '.' - CLS CLUSTER Record (Type 19)
    12 - 'M' or '.' - MSC MSCP Record (Type 21)
    13 - 'T' or '.' - TRN TRANSACTION Record (Type 22)
```

A DEMSPR file observation having at least one record present of each record type would have a SPRMASK equal to 'SPIFPLDFDSCMT'. A record type that is not present is denoted with a period.

TYPE: Retained data element - Character, length 13.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SPRMASK is constructed when all of the available VMS MONITOR records for the DEMSPR file are merged together by the CA MICS module DEXMON.

RANGE OF VALUE: SPRMASK contains any valid combination of the byte values shown above.

SUMMARIZATION PROCESS: SPRMASK is meaningful only at the DETAIL level of summarization.

USAGE NOTES: SPRMASK provides you with an audit trail to identify which VMS MONITOR record types were used to build observations in the CA MICS DEMSPR file. This element can be useful in analyzing the completeness of the SPR data at the DETAIL level. Examining SPRMASK reveals why certain fields may be missing.

```
+-----------------+
| S P R M D P G   |  Modified Page Count
+-----------------+
```

SPRMDPG contains the count of pages on the modified page list at the end of DURATION (measurement interval).

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMNMP - Minimum Modified Page Count
SPRMDPGC - PGE Modified Page Count
SPRMODP - IO Modified Page Count

ELEMENT DERIVATION: SPRMDPG is set to the value of the MNR_SYS$L_MFYCNT field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE: SPRMDPG takes an integer value from 0 to the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS: SPRMDPG contains the last value of SPRMDPG processed in the level of summarization.

USAGE NOTES: Modified pages are those that have been written to the page file by the memory manager but have not been needed for another process yet. SPRMDPG is the number of pages that have been written out but not yet needed by another process. They are kept on the modified list until a page fault occurs that recalls them to the process they came from or free page list depletion requires them to be moved to the free page list.

```
+-----------------+
| S P R M D P G C |  PGE Modified Page Count
+-----------------+
```

SPRMDPGC contains the count of pages on the modified page list at the end of DURATION (measurement interval).

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMNPMP - PGE Minimum Modified Page Count
SPRMPGFL - Modified List Page Faults
SPRMODP - IO Modified Page Count
SPRMDPG - Modified Page Count

ELEMENT DERIVATION: SPRMDPGC is set to the value of the MNR_PAG$L_MFYCNT field of the VMS MONITOR PAGE record.

RANGE OF VALUE: SPRMDPGC takes any integer value from 0 to the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS: SPRMDPGC contains the last value of SPRMDPGC processed in the level of summarization.

USAGE NOTES: Modified pages are those that have been written to the page file by the memory manager but have not been needed for another process yet. SPRMDPGC is the number of pages that have been written out but not yet needed by another process. The pages are kept on the modified list

until a page fault occurs that recalls them to the process from which they came or free page list depletion requires them to be moved to the free page list.

| S P R M N F P | Minimum Free Page Count

SPRMNFP contains the minimum count of pages on the free page list during DURATION (measurement interval).

TYPE: Minimum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFRPG - Free Page Count
SPRMNPFP - Minimum Free Page Count
SPRMNIFP - Minimum Free Page Count

ELEMENT DERIVATION: SPRMNFP is set to the minimum value of the MNR_SYS$L_FREECNT field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE: SPRMNFP takes an integer value from 0 to the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS: SPRMNFP contains the minimum value of SPRMNFP processed for the level of summarization.

USAGE NOTES: SPRMNFP represents the smallest number of 512-byte pages of physical memory that was available for processes to allocate. If SPRMNFP is smaller than FREELIM, the memory manager was invoked to increase the free page count to FREEGOAL. FREELIM and FREEGOAL are SYSGEN parameters. If SPRMNFP is less than FREELIM, there is probably not enough physical memory on the system.

| S P R M N F R E | Minimum Free Page List Size

SPRMNFRE contains the minimum count of pages on the free page list during DURATION (measurement interval).

TYPE: Minimum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFRE - Free Page Count
SPRMNPFP - Minimum Free Page Count
SPRMNIFP - Minimum Free Page Count
SPRMNFP - Minimum Free Page Count

ELEMENT DERIVATION: SPRMNFRE is set to the minimum value of the MNR_CLU$L_FRLIST field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRMNFRE takes an integer value from 0 to the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS: SPRMNFRE contains the minimum value of SPRMNFRE processed for the level of summarization.

USAGE NOTES: SPRMNFRE represents the smallest number of 512-byte pages of physical memory that was available for processes to allocate. If SPRMNFRE is smaller than FREELIM, the memory manager was invoked to increase the free page count to FREEGOAL. FREELIM and FREEGOAL are SYSGEN parameters. If SPRMNFRE is less than FREELIM, there is probably not enough physical memory on the system.

| S P R M N H O L | Minimum Unused KBytes

SPRMNHOL is the size (in bytes) of the smallest unused space in the dynamic nonpaged pool.

TYPE: Minimum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRHOLES - Nonpaged Pool Unused KBytes

ELEMENT DERIVATION: SPRMNHOL is set to the value of the MNR_POO$L_HOLESUM field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRMNHOL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMNHOL contains the smallest value of SPRMNHOL processed for the level of summarization.

USAGE NOTES: As the nonpaged pool is used, areas that were used and then freed are called holes. The sum of the sizes of the holes, measured in kilobytes, is stored in SPRHOLES. When the nonpaged pool is depleted, memory allocations must come from paged memory, which could negatively affect performance for processes that use the resource extensively. SPRMNHOL is the smallest amount of space freed in the level of summarization.

```
+-----------------+
| S P R M N I F P |  IO Minimum Free Page Count
+-----------------+
SPRMNIFP contains the minimum count of pages on the free page
list during DURATION (measurement interval).

TYPE:  Minimum - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRFREEP - Free Page Count
           SPRMNPFP - Minimum Free Page Count
           SPRMNFP  - Minimum Free Page Count

ELEMENT DERIVATION:  SPRMNIFP is set to the minimum value of
the MNR_IO$L_FREECNT field of the VMS MONITOR IO record.

RANGE OF VALUE:  SPRMNIFP takes an integer value from 0 to
the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS:  SPRMNIFP contains the minimum value
of SPRMNIFP processed for the level of summarization.

USAGE NOTES:  SPRMNIFP represents the smallest number of 512-
byte pages of physical memory that was available for
processes to allocate.  If SPRMNIFP is smaller than FREELIM,
the memory manager was invoked to increase the free page
count to FREEGOAL.  FREELIM and FREEGOAL are SYSGEN
parameters.  If SPRMNIFP is less than FREELIM, there is
probably not enough physical memory on the system.


+-----------------+
| S P R M N I M P |  IO Minimum Modified Page Count
+-----------------+
SPRMNIMP contains the minimum count of pages on the modified
page list during DURATION (measurement interval).

TYPE:  Minimum - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRMODP - IO Modified Page Count
           SPRMNPMP - PGE Minimum Modified Page Count
           SPRMNMP  - Minimum Modified Page Count

ELEMENT DERIVATION:  SPRMNIMP is set to the minimum value of
the MNR_IO$L_MFYCNT field of the VMS MONITOR IO record.

RANGE OF VALUE:  SPRMNIMP takes an integer value from 0 to
the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS:  SPRMNIMP contains the minimum value
of SPRMNIMP processed for the level of summarization.

USAGE NOTES:  Modified pages are those that have been written
to the page file by the memory manager but have not been
needed for another process yet.  SPRMNIMP is the smallest
number of pages that have been written out but not yet needed
by another process or for the free page list.  If both
SPRMNIMP and SPRMNIFP are small, the system may need
additional physical memory.


+-----------------+
| S P R M N I P F |  Minimum Intermediate Req Packets Free
+-----------------+
SPRMNIPF contains the minimum number of intermediate request
packets (IRPs) free.

TYPE:  Minimum - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRIPFRE - Intermediate Request Packets Free

ELEMENT DERIVATION:  SPRMNIPF is set to the value of the
MNR_POO$L_SRPCNT field of the VMS MONITOR POOL record.

RANGE OF VALUE:  SPRMNIPF takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRMNIPF contains the minimum value
of SPRMNIPF processed for the level of summarization.

USAGE NOTES:  The SYSGEN parameter IRPCOUNT determines how
many IRPs are preallocated for use by processes on the
system.  When a process requests pool space that fits into an
IRP, one is allocated to the process.  SPRMNIPF is the lowest
number of IRPs available for use during the DURATION
(measurement interval).


+-----------------+
| S P R M N L B K |  Minimum Largest Block
+-----------------+
SPRMNLBK contains the minimum size of the largest block of
unused space in the dynamic nonpaged pool.

TYPE:  Minimum - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRBIGHL - Largest Block Unused
```

ELEMENT DERIVATION: SPRMNLBK is set to the value of the MNR_POO$L_BIGHOLE field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRMNLBK takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMNLBK contains the minimum value of SPRMNLBK processed for the level of summarization.

USAGE NOTES: As the nonpaged pool is used, areas that were used and then freed are called holes. The size of the largest hole is stored in SPRBIGHL. When dynamic memory is required, if the size is larger than SPRBIGHL, the memory manager knows that it can not serve that much contiguous space to the process. If SPRBIGHL is too small, the dynamic portion of the pool is being fragmented badly. If this is the case, the applications should be reviewed for their use of pool. SPRMNLBK is the smallest value recorded in SPRBIGHL.

| S P R M N L P F | Minimum Large Request Packets Free

SPRMNLPF contains the minimum number of large request packets (LRPs) free.

TYPE: Minimum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRLPFRE - Large Request Packets Free

ELEMENT DERIVATION: SPRMNLPF is set to the value of the MNR_POO$L_LRPCNT field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRMNLPF takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMNLPF contains the minimum value of SPRMNLPF processed for the level of summarization.

USAGE NOTES: The SYSGEN parameter LRPCOUNT determines how many (LRPs) are preallocated for use by processes on the system. When a process requests pool space that fits into an LRP, one is allocated to the process. SPRMNLPF is the lowest number of LRPs available for use during the DURATION (measurement interval).

| S P R M N L P K | Minimum Large Request Packets Left

SPRMNLPK contains the minimum number of LRPs left.

This element will have missing values for systems running VMS Version 6.0 or later.

TYPE: Minimum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRLRGPK - Large Request Packets Left

ELEMENT DERIVATION: SPRMNLPK is set to the value of the MNR_NET$L_LRPCNT field of the VMS MONITOR DECNET record.

RANGE OF VALUE: SPRMNLPK takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMNLPK contains the minimum value of SPRMNLPK processed for the level of summarization.

USAGE NOTES: LRPs are preallocated portions of the nonpaged pool. LRPs are primarily used in DECnet communications to store incoming DECnet packets. When LRPs are depleted, VMS adds more LRPs to the pool. The memory used for the new LRPs is taken from user memory. Messages received while new LRPs are being added will be retransmitted. SPRMNLPK reports the smallest number that the LRP count became.

| S P R M N M P | Minimum Modified Page Count

SPRMNMP contains the minimum count of pages on the modified page list during DURATION (measurement interval).

TYPE: Minimum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMDPG - Modified Page Count
SPRMNPMP - PGE Minimum Modified Page Count
SPRMNIMP - IO Minimum Modified Page Count

ELEMENT DERIVATION: SPRMNMP is set to the minimum value of the MNR_SYS$L_MFYCNT field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE: SPRMNMP takes an integer value from 0 to the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS: SPRMNMP contains the minimum value of

SPRMNMP processed for the level of summarization.

USAGE NOTES: Modified pages are those that have been written to the page file by the memory manager but have not been needed for another process yet. SPRMNMP is the smallest number of pages that have been written out but not yet needed by another process or for the free page list. If both SPRMNMP and SPRMNFP are small, the system may need additional physical memory.

S P R M N P F P — PGE Minimum Free Page Count

SPRMNPFP contains the minimum count of pages on the free page list during DURATION (measurement interval).

TYPE: Minimum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFRPGC - PGE Free Page Count

ELEMENT DERIVATION: SPRMNPFP is set to the smallest value recorded in the MNR_PAG$L_FREECNT field of the VMS MONITOR PAGE record.

RANGE OF VALUE: SPRMNPFP takes any integer value from 0 to the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS: SPRMNPFP contains the minimum value of SPRMNPFP processed for the level of summarization.

USAGE NOTES: SPRMNPFP represents the smallest number of 512-byte pages of physical memory that was available for processes to allocate. If SPRMNPFP is smaller than FREELIM, the memory manager was invoked to increase the free page count to FREEGOAL. FREELIM and FREEGOAL are SYSGEN parameters. If SPRMNPFP is less than FREELIM, there is probably not enough physical memory on the system.

S P R M N P M P — PGE Minimum Modified Page Count

SPRMNPMP contains the minimum count of pages on the modified page list during DURATION (measurement interval).

TYPE: Minimum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMDPGC - PGE Modified Page Count
SPRMNIMP - IO Minimum Modified Page Count
SPRMNMP - Minimum Modified Page Count

ELEMENT DERIVATION: SPRMNPMP is set to the smallest value of the MNR_PAG$L_MFYCNT field of the VMS MONITOR PAGE record.

RANGE OF VALUE: SPRMNPMP takes any integer value from 0 to the number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS: SPRMNPMP contains the minimum value of SPRMNPMP processed for the level of summarization.

USAGE NOTES: Modified pages are those that have been written to the page file by the memory manager but have not been needed for another process yet. SPRMNPMP is the smallest number of pages that have been written out to but not yet needed by another process or for the free page list. If both SPRMNPMP and SPRMNPFP are small, the system may need additional physical memory.

S P R M N S B K — Minimum Smallest Block

SPRMNSBK contains the number of smallest block.

TYPE: Minimum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRSMHL - Smallest Block Unused

ELEMENT DERIVATION: SPRMNSBK is set to the value of the MNR_POO$L_SMALLHOLE field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRMNSBK takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMNSBK contains the minimum value of SPRMNSBK processed for the level of summarization.

USAGE NOTES: As the nonpaged pool is used, areas that were used and then freed are called holes. The size of the smallest hole is stored in SPRSMHL. When dynamic memory is required, if the size is less than SPRSMHL, the memory manager allocates the space from that hole. SPRSMHL is measured in bytes. SPRMNSBK is the smallest value stored in SPRSMHL.

```
+-----------------+
| S P R M N S P F |  Minimum Small Request Packets Free
+-----------------+
SPRMNSPF contains the minimum number of small request packets
(SRPs) free.

TYPE:  Minimum - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRSPFRE - Small Request Packets Free

ELEMENT DERIVATION:  SPRMNSPF is set to the value of the
MNR_POO$L_SRPCNT field of the VMS MONITOR POOL record.

RANGE OF VALUE:  SPRMNSPF takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRMNSPF contains the minimum value
of SPRMNSPF processed for the level of summarization.

USAGE NOTES:  The SYSGEN parameter SRPCOUNT determines how
many SRPs are preallocated for use by processes on the
system.  When a process requests pool space that fits into an
SRP, one is allocated to the process.  SPRMNSPF is the
minimum number of SRPs that were available for use.


+-----------------+
| S P R M N U C N |  Minimum Unused Contiguous Space
+-----------------+
SPRMNUCN contains the minimum amount of unused contiguous
space.

TYPE:  Minimum - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRUNCON - Nonpaged Pool Unused Contiguous Space

ELEMENT DERIVATION:  SPRMNUCN is set to the value of the
MNR_POO$L_HOLECNT field of the VMS MONITOR POOL record.

RANGE OF VALUE:  SPRMNUCN takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRMNUCN contains the minimum value
of SPRMNUCN processed for the level of summarization.

USAGE NOTES:  As the dynamic pool becomes fragmented,
SPRUNCON keeps track of the amount of free contiguous space.
If SPRUNCON is growing, the pool can badly fragment and
degrade system performance.  The smallest value reached by
SPRUNCON is stored in SPRMNUCN.


+-----------------+
| S P R M O D P   |  IO Modified Page Count
+-----------------+
SPRMODP contains the count of pages on the modified page list
at the end of DURATION (measurement interval).

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRMNIMP - IO Minimum Modified Page Count
           SPRMDPGC - PGE Modified Page Count
           SPRMDPG  - Modified Page Count

ELEMENT DERIVATION:  SPRMODP is set to the value of the
MNR_IO$L_MFYCNT field of the VMS MONITOR IO record.

RANGE OF VALUE:  SPRMODP takes an integer value from 0 to the
number of 512-byte pages in physical memory.

SUMMARIZATION PROCESS:  SPRMODP contains the last value of
SPRMODP processed in the level of summarization.

USAGE NOTES:  Modified pages are those that have been written
to the page file by the memory manager but have not been
needed for another process yet.  SPRMODP is the number of
pages that have been written out but not yet needed by
another process.  They are kept on the modified list until a
page fault occurs that recalls them to the process to which
they belong or until free page list depletion requires them
to be moved to the free page list.


+-----------------+
| S P R M P G F L |  Modified List Page Faults
+-----------------+
SPRMPGFL contains the number of pages allocated from the
modified page list in DURATION (measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSMPL - Modified List Page Fault Rate

ELEMENT DERIVATION:  SPRMPGFL is set to the value of the
MNR_PAG$L_MFYFLTS field of the VMS MONITOR PAGE record.

RANGE OF VALUE:  SPRMPGFL takes any integer value from 0 to
```

2,147,483,647.

SUMMARIZATION PROCESS: SPRMPGFL contains the sum of all values of SPRMPGFL processed in the level of summarization.

USAGE NOTES: Modified pages are those that have been written to the page file by the memory manager but have not been needed for another process yet. SPRMPGFL is the number of times that those marked pages were requested again by the process before they were written to the page file.

Modified page list faults are among the best kind of page faults to have because no I/O is required, just memory mapping and status changing. The system tries to keep as many modified pages on the list as it can, but if the free list is depleted it takes modified pages before it begins to trim working sets or to swap processes.

S P R M S C C T — MSC MSCP SERVER Record Count (Type 21)

SPRMSCCT contains a count of VMS MONITOR MSCP_SERVER records processed by the CA MICS VAX/VMS Analyzer.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS MONITOR MSCP_SERVER record is stored, SPRMSCCT is set to 1.

RANGE OF VALUE: In the DETAIL timespan of the CA MICS VMS System Profile file, SPRMSCCT is always 0 or 1. At higher levels of summarization, its value ranges from 0 to the total number of VMS MONITOR MSC records processed in the level of summarization.

SUMMARIZATION PROCESS: SPRMSCCT contains the sum of all values of SPRMSCCT processed in the level of summarization.

USAGE NOTES: SPRMSCCT is used as a divisor for calculating averages and ratios in the level of summarization.

S P R M S C D R — MSC MSCP SERVER Record Duration (Type 21)

SPRMSCDR contains the number of seconds in the "reporting interval" of a given VMS MONITOR record. Reporting interval is the period of time during which a monitor gathers and reports statistics on this record.

TYPE: Common accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: SPRMSCCT - MSC MSCP SERVER Record Count (Type 21)
INTERVLS - Number of Recording Intervals
STARTTS - Start Time Stamp
ENDTS - End Time Stamp
DEXDUR - Requested Recording Interval Time

ELEMENT DERIVATION: SPRMSCDR contains the difference between the time stamp of the current VMS MONITOR record and the time stamp of the previously processed record of the same type.

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SPRMSCDR contains the sum of all occurrences of SPRMSCDR that were included in the level of summarization.

USAGE NOTES: Note that SPRMSCDR is uniquely calculated for this VMS MONITOR record type before merging records together into the DEMSPR file.

S P R M S C R D — MSCP Reads

SPRMSCRD contains the count of requests for read I/Os by remote processes.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFCPRI - FCP Disk Read I/Os

ELEMENT DERIVATION: SPRMSCRD is set to the value of the MNR_MSC$L_ field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRMSCRD takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMSCRD contains the sum of all values of SPRMSCRD processed in the level of summarization.

USAGE NOTES: SPRMSCRD counts only the requests that come from remote processes. Because the MSCP server runs as a local process, the I/Os it generates are also counted in the normal I/O processing of the recording node. The actual number of reads done by the MSCP server is SPRMSCRD plus the number of fragments caused by read requests. Because there is no way of separating read fragments and write fragments, knowledge of the typical I/O request size by the applications on the remote node is needed for good estimates.

S P R M S C S P | MSCP Splits

SPRMSCSP contains the count of requests that were split into fragments so they could be served.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFRGMT - MSCP Fragments

ELEMENT DERIVATION: SPRMSCSP is set to the value of the MNR_MSC$L_SPLIT field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRMSCSP takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMSCSP contains the sum of all values of SPRMSCSP processed in the level of summarization.

USAGE NOTES: When MSCP server requests are too large to be served by the buffers, the server splits the requests into smaller fragments. For example, if the current request is for 60 blocks but the buffers can only handle 16-block requests, the current request will be split into four separate fragments. Three fragments will contain 16 blocks each and one fragment will contain 14 blocks. SPRMSCSP will then be incremented by 1. Splitting takes more CPU time and I/O transactions than if the request can be served in one I/O transaction. If SPRMSCSP is large, investigate the option of enlarging the MSCP buffers.

S P R M S C W R | MSCP Writes

SPRMSCWR contains the count of requests for write I/Os by remote processes.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFCPWI - FCP Disk Write I/Os

ELEMENT DERIVATION: SPRMSCWR is set to the value of the MNR_MSC$L_WRITE field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRMSCWR takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMSCWR contains the sum of all values of SPRMSCWR processed in the level of summarization.

USAGE NOTES: SPRMSCWR counts the requests that come from remote processes. Because the MSCP server runs as a local process, the I/Os it generates are also counted in the normal I/O processing of the recording node. The actual number of writes by the MSCP server is SPRMSCWR plus the number of fragments caused by write requests. Because there is no way of separating read fragments and write fragments, knowledge of the typical I/O request size by the applications on the remote node is needed for good estimates.

S P R M S T A B | Max Transaction Abort Rate

SPRMSTAB contains the maximum count per second of transaction operations aborted by DECdtm services. It is the total of planned and unplanned aborts.

TYPE: Maximum - Numeric, measurement unit is integer.

SEE ALSO: SPRPSTAB - Transaction Abort Rate
SPRCNTAB - Transactions Aborted Count

ELEMENT DERIVATION: SPRMSTAB is calculated in the detail time span as: SPRMSTAB = SPRCNTAB / DURATION . In the detail time span this calculation represents a single observation over the entire interval and is thus an average rate (and a single measurement). At summarized levels, this element is set to the maximum value for this interval average.

RANGE OF VALUE: SPRMSTAB ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRMSTAB contains the maximum of all values of SPRMSTAB processed in the level of summarization.

USAGE NOTES: This element shows the highest sustained rate for the level of summarization. This measurement does not account for higher instantaneous rates as it is the maximum of an average over the original recording interval.

S P R M S T A D — Max Transaction Remote Add Rate

SPRMSTAD contains the maximum count per second of add remote (to a remote subordinate parent) branch operations by DECdtm services.

TYPE: Maximum - Numeric, measurement unit is integer.

SEE ALSO: SPRPSTAD - Transaction Remote Add Rate
SPRCNTAD - Add Remote Branch Ops Count

ELEMENT DERIVATION: SPRMSTAD is calculated in the detail time span as: SPRMSTAD = SPRCNTAD / DURATION . In the detail time span this calculation represents a single observation over the entire interval and is thus an average rate (and a single measurement). At summarized levels, this element is set to the maximum value for this interval average.

RANGE OF VALUE: SPRMSTAD ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRMSTAD contains the maximum of all values of SPRMSTAD processed in the level of summarization.

USAGE NOTES: This element shows the highest sustained rate for the level of summarization. This measurement does not account for higher instantaneous rates as it is the maximum of an average over the original recording interval.

S P R M S T B R — Max Transaction Remote Start Rate

SPRMSTBR contains the maximum count per second of start remote (to a remote parent) branch operations by DECdtm services.

TYPE: Maximum - Numeric, measurement unit is integer.

SEE ALSO: SPRPSTBR - Transaction Remote Start Rate
SPRCNTBR - Start Remote Branch Ops Count

ELEMENT DERIVATION: SPRMSTBR is calculated in the detail time span as: SPRMSTBR = SPRCNTBR / DURATION . In the detail time span this calculation represents a single observation over the entire interval and is thus an average rate (and a single measurement). At summarized levels, this element is set to the maximum value for this interval average.

RANGE OF VALUE: SPRMSTBR ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRMSTBR contains the maximum of all values of SPRMSTBR processed in the level of summarization.

USAGE NOTES: This element shows the highest sustained rate for the level of summarization. This measurement does not account for higher instantaneous rates as it is the maximum of an average over the original recording interval.

S P R M S T C O — Max Transaction Total Commit Rate

SPRMSTCO contains the maximum count per second of transactions committed in total. It is based on the combined count of one-phase and two-phase commit operations completed by DECdtm services.

TYPE: Maximum - Numeric, measurement unit is integer.

SEE ALSO: SPRPSTCO - Transaction Total Commit Rate
SPRCNTCO - Transactions Total Committed Count

ELEMENT DERIVATION: SPRMSTCO is calculated in the detail time span as: SPRMSTCO = SPRCNTCO / DURATION . In the detail time span this calculation represents a single observation over the entire interval and is thus an average rate (and a single measurement). At summarized levels, this element is set to the maximum value for this interval average.

RANGE OF VALUE: SPRMSTCO ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRMSTCO contains the maximum of all values of SPRMSTCO processed in the level of summarization.

USAGE NOTES: This element shows the highest sustained rate

for the level of summarization. This measurement does not account for higher instantaneous rates as it is the maximum of an average over the original recording interval.

| S P R M S T E N | Max Transaction End Rate

SPRMSTEN contains the maximum count per second of transaction operations ended by DECdtm services.

TYPE: Maximum - Numeric, measurement unit is integer.

SEE ALSO: SPRPSTEN - Transaction End Rate
SPRCNTEN - Transactions Ended Count

ELEMENT DERIVATION: SPRMSTEN is calculated in the detail time span as: SPRMSTEN = SPRCNTEN / DURATION . In the detail time span this calculation represents a single observation over the entire interval and is thus an average rate (and a single measurement). At summarized levels, this element is set to the maximum value for this interval average.

RANGE OF VALUE: SPRMSTEN ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRMSTEN contains the maximum of all values of SPRMSTEN processed in the level of summarization.

USAGE NOTES: This element shows the highest sustained rate for the level of summarization. This measurement does not account for higher instantaneous rates as it is the maximum of an average over the original recording interval.

| S P R M S T O P | Max Transaction One-Phase Init Rate

SPRMSTOP contains the maximum count per second of one-phase commit operations initiated by DECdtm services.

TYPE: Maximum - Numeric, measurement unit is integer.

SEE ALSO: SPRPSTOP - Transaction One-Phase Init Rate
SPRCNTOP - One-Phase Commits Initiated

ELEMENT DERIVATION: SPRMSTOP is calculated in the detail time span as: SPRMSTOP = SPRCNTOP / DURATION . In the detail time span this calculation represents a single observation over the entire interval and is thus an average rate (and a single measurement). At summarized levels, this element is set to the maximum value for this interval average.

RANGE OF VALUE: SPRMSTOP ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRMSTOP contains the maximum of all values of SPRMSTOP processed in the level of summarization.

USAGE NOTES: This element shows the highest sustained rate for the level of summarization. This measurement does not account for higher instantaneous rates as it is the maximum of an average over the original recording interval.

| S P R M S T P R | Max Transaction Prepare Rate

SPRMSTPR contains the maximum count per second of transactions placed in the prepare state by DECdtm services.

TYPE: Maximum - Numeric, measurement unit is integer.

SEE ALSO: SPRPSTPR - Transaction Prepare Rate
SPRCNTPR - Transactions Prepared Count

ELEMENT DERIVATION: SPRMSTPR is calculated in the detail time span as: SPRMSTPR = SPRCNTPR / DURATION . In the detail time span this calculation represents a single observation over the entire interval and is thus an average rate (and a single measurement). At summarized levels, this element is set to the maximum value for this interval average.

RANGE OF VALUE: SPRMSTPR ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRMSTPR contains the maximum of all values of SPRMSTPR processed in the level of summarization.

USAGE NOTES: This element shows the highest sustained rate for the level of summarization. This measurement does not account for higher instantaneous rates as it is the maximum of an average over the original recording interval.

```
+-----------------+
| S P R M S T S T |  Max Transaction Start Rate
+-----------------+
SPRMSTST contains the maximum count per second of transaction
operations started by DECdtm services.

TYPE:  Maximum - Numeric, measurement unit is integer.

SEE ALSO:  SPRPSTST - Transaction Start Rate
           SPRCNTST - Transactions Started Count

ELEMENT DERIVATION:  SPRMSTST is calculated in the detail
time span as:  SPRMSTST = SPRCNTST / DURATION . In the detail
time span this calculation represents a single observation
over the entire interval and is thus an average rate (and a
single measurement).  At summarized levels, this element is
set to the maximum value for this interval average.

RANGE OF VALUE:  SPRMSTST ranges from a minimum of 0 to a
maximum limited only by practical bounds.

SUMMARIZATION PROCESS:  SPRMSTST contains the maximum of all
values of SPRMSTST processed in the level of summarization.

USAGE NOTES:  This element shows the highest sustained rate
for the level of summarization.  This measurement does not
account for higher instantaneous rates as it is the maximum
of an average over the original recording interval.


+-----------------+
| S P R M W A I T |  Miscellaneous Resource Wait
+-----------------+
SPRMWAIT contains the count of processes waiting on resources
other than paging lists, event flags, timers, or CPU access.

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  PROSTATE - State
           SPRMXMRW - Maximum Miscellaneous Resource Wait

ELEMENT DERIVATION:  SPRMWAIT is set to the value of the
MNR_STA$L_MWAIT field from the VMS MONITOR STATES record.

RANGE OF VALUE:  SPRMWAIT takes any integer value from 0 to
the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS:  SPRMWAIT contains the last value of
SPRMWAIT processed in the level of summarization.

USAGE NOTES:  The processes counted in SPRMWAIT are waiting
for resources that ordinarily should not be busy.  SPRMWAIT
represents processes that are involuntarily forced to wait
for system-wide resources such as mailboxes, asynchronous
trap mechanisms, and swap and page files.  If SPRMWAIT is
consistently greater than zero, then the VMS system
parameters and memory management files should be examined for
areas of improvement.  The areas to look at first are the
size of the page and swap files and the number of I/Os on the
disks where those files reside.


+-----------------+
| S P R M X C E F |  Maximum Common Event Flag Wait
+-----------------+
SPRMXCEF contains the maximum count of processes waiting for
a common event flag to be signaled.

TYPE:  Maximum - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRCEF - Common Event Flag Wait

ELEMENT DERIVATION:  SPRMXCEF is set to the maximum value
reached by the MNR_STA$L_CEF field from the VMS MONITOR
STATES record during the sampling period.

RANGE OF VALUE:  SPRMXCEF takes any integer value from 0 to
the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS:  SPRMXCEF contains the maximum value
of SPRMXCEF processed for the level of summarization.

USAGE NOTES:  SPRMXCEF records the maximum number of
processes that were waiting for resources that are in global
memory or are waiting on a system-wide event.  If the number
in SPRMXCEF represents a large percentage of the processes on
the system, the system is probably locking too many global
resources.  These resources represent a bottleneck that may
eventually degrade system performance.


+-----------------+
| S P R M X C L K |  Maximum Current Locks
+-----------------+
SPRMXCLK contains the maximum number of current locks held
during DURATION (measurement interval).

TYPE:  Maximum - Numeric, measurement unit is integer, 4
bytes long.
```

SEE ALSO: SPRCURLK - Current Locks

ELEMENT DERIVATION: SPRMXCLK is set to the value of the MNR_LCK$L_NUMLOCKS field of the VMS MONITOR LOCK record.

RANGE OF VALUE: SPRMXCLK takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMXCLK contains the maximum value of SPRMXCLK processed for the level of summarization.

USAGE NOTES: SPRCURLK counts the number of resources on the system that have been exclusively granted to a process. SPRMXCLK is the maximum value that number reached.

S P R M X C P — Maximum Current Processes

SPRMXCP contains the maximum number of current processes on the VMS system.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRCUR - Current Processes

ELEMENT DERIVATION: SPRMXCP is set to the maximum value reached by the MNR_STA$L_CUR field of the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXCP takes any integer value from 1 to the number of processors installed in the VAX hardware.

SUMMARIZATION PROCESS: SPRMXCP contains the maximum value of SPRMXCP processed for the level of summarization.

USAGE NOTES: In a single processor VAX system, SPRMXCP will always be 1--the process running VMS MONITOR itself. In multiprocessor systems, SPRMXCP can be as high as the number of processors installed in the hardware. If the value in SPRMXCP is less than the number of processors, then the workload is never sufficient to make all of the processors busy at the same time.

S P R M X C P W — Maximum Collided Page Wait

SPRMXCPW contains the maximum count of processes waiting for pages of memory that have been placed in the system page file.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRCOLPG - Collided Page Wait

ELEMENT DERIVATION: SPRMXCPW is set to the maximum value reached by the MNR_STA$L_COLPG field of the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXCPW takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXCPW contains the maximum value of SPRMXCPW processed for the level of summarization.

USAGE NOTES: A high value in SPRMXCPW indicates that the process working set sizes are not set high enough. If there is still memory on the free page list, the process working sets should be increased to help lower collided paging. Collided paging occurs when a process requests access to a page of memory that had been in memory at one time but has been written to the page file to free memory to satisfy other memory accesses.

If SPRMXCPW is high and there is no memory on the free page list, the system may need additional physical memory. If SPRMXCPW is high, the free page list is nearly empty, and only a few processes are outswapped, the system is probably tuned as well as it can be for the limited resources available.

A high value in SPRMXCPW will have negative effects on the CPU and the I/O subsystem. The CPU usage will not show in any of the processes on the system but will be recorded as kernel and executive mode processing.

S P R M X C R S — Maximum Current Resources

SPRMXCRS contains the maximum number of current resources.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRCURRS - Current Resources

ELEMENT DERIVATION: SPRMXCRS is set to the value of the MNR_LCK$L_NUMRES field of the VMS MONITOR LOCK record.

RANGE OF VALUE: SPRMXCRS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMXCRS contains the maximum value of SPRMXCRS processed for the level of summarization.

USAGE NOTES: SPRCURRS counts the current number of enqueue control blocks registered with the Lock Manager. When a process wants exclusive access to a system-wide resource, it requests an enqueue control black associated with the resource. The resource is then counted as both a current resource and as a current lock (SPRCURLK). When the process is done with exclusive use of the resource, it dequeues the resource, removing it from the list of current locks. The enqueue control block remains associated with the resource until the control block is deregistered. SPRMXCRS is the largest value reached by SPRCURRS.

S P R M X C S I — Maximum Compute State Wait, Inswapped

SPRMXCSI contains the maximum count of processes that are ready to compute but are waiting to be scheduled.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRCOM - Compute State Wait, Inswapped

ELEMENT DERIVATION: SPRMXCSI is set to the maximum value reached by the MNR_STA$L_COM field from the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXCSI takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXCSI contains the maximum value of SPRMXCSI processed for the level of summarization.

USAGE NOTES: SPRMXCSI records the maximum number of processes that were ready to compute but were prevented from doing so because the CPU was busy with higher priority tasks. If the number in SPRMXCSI is a significant percentage of the total number of processes on the system, the CPU is not fast enough for the workload.

S P R M X C S O — Maximum Compute State Wait, Outswapped

SPRMXCSO contains the maximum count of processes that are ready to compute, are waiting to be scheduled, and are swapped out of memory.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRCOMO - Compute State Wait, Outswapped

ELEMENT DERIVATION: SPRMXCSO is set to the maximum value reached by the MNR_STA$L_COMO field from the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXCSO takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXCSO contains the maximum value of SPRMXCSO processed for the level of summarization.

USAGE NOTES: SPRMXCSO records the maximum number of processes that were ready to compute but that were prevented from doing so because the scheduler swapped them out before the CPU was free to process them. If SPRMXCSO is nonzero, the system probably needs improvements. The most common reason for a nonzero value in SPRMXCSO is that there is not enough physical memory. A second reason is that the CPU is not fast enough for the workload, but that may be due to excessive paging and swapping.

S P R M X F P W — Maximum Free Page Wait

SPRMXFPW contains the maximum count of processes waiting for pages of memory from the free page list.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRFPGWT - Free Page Wait

ELEMENT DERIVATION: SPRMXFPW is set to the maximum value reached by the MNR_STA$L_FPG field of the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXFPW takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXFPW contains the maximum value of SPRMXFPW processed for the level of summarization.

USAGE NOTES: A nonzero value in SPRMXFPW means that at some point, some processes were waiting for new pages from the free page list but none were available. SPRMXFPW usually will not get very large unless processes activate many images and memory is constrained. Increasing the free list cache size in the SYSGEN parameters should decrease the number of processes that must wait for free pages.

S P R M X H W I — Maximum Hibernate Wait, Inswapped

SPRMXHWI contains the maximum count of processes that signaled the operating system that they will not process for some period of time.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRHIB - Hibernate Wait, Inswapped

ELEMENT DERIVATION: SPRMXHWI is set to the maximum value reached by the MNR_STA$L_HIB field from the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXHWI takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXHWI contains the maximum value of SPRMXHWI processed for the level of summarization.

USAGE NOTES: The count in SPRMXHWI does not affect VMS performance to any great extent. If the count is large and the maximum number of processes allowed on the system is reached, then processes that are hibernating take up process slots without doing any work at that time. Hibernating processes will usually do useful work when they are awakened by the system, so it is still important to have hibernating processes remain on the system.

S P R M X H W O — Maximum Hibernate Wait, Outswapped

SPRMXHWO contains the maximum count of processes that signaled the operating system that they will not process for some period of time and were then swapped out of memory.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRHIBO - Hibernate Wait, Outswapped

ELEMENT DERIVATION: SPRMXHWO is set to the maximum value reached by the MNR_STA$L_HIBO field from the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXHWO takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXHWO contains the maximum value of SPRMXHWO processed for the level of summarization.

USAGE NOTES: The count in SPRMXHWO does not affect VMS performance to any great extent. If the count is large and the maximum number of processes allowed on the system is reached, then processes that are hibernating take up process slots without doing any work at that time. Hibernating processes usually will do useful work when they are awakened by the system, so it is still important to have hibernating processes remain on the system.

If there are any outswapped processes on the system, it is best that they be in the hibernate or the suspended state. In those states swapping has the least effect on performance.

S P R M X I P U — Maximum Intermediate Req Packet in Use

SPRMXIPU contains the maximum number of intermediate request packets (IRPs) in use.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRIPUSE - Intermediate Request Packets in Use

ELEMENT DERIVATION: SPRMXIPU is set to the value of the MNR_POO$L_IRPINUSE field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRMXIPU takes any integer value from 0 to

2,147,483,647.

SUMMARIZATION PROCESS: SPRMXIPU contains the maximum value of SPRMXIPU processed for the level of summarization.

USAGE NOTES: The SYSGEN parameter IRPCOUNT determines how many IRPs are preallocated for use by processes on the system. When a process requests pool space that fits into an IRP, one is allocated to the process. SPRMXIPU is the maximum count of IRPs in use.

S P R M X K B U — Maximum Nonpaged Pool KBytes in Use

SPRMXKBU contains the maximum number of kilobytes in use.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRKBUSE - Nonpaged Pool KBytes in Use

ELEMENT DERIVATION: SPRMXKBU is set to the value of the MNR_POO$L_DYNINUSE field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRMXKBU takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMXKBU contains the maximum value of SPRMXKBU processed for the level of summarization.

USAGE NOTES: The dynamic portion of the nonpaged pool serves requests that do not closely fit the preallocated packets. It is also used as an overflow when the packets are in use. SPRKBUSE is the total amount, in kilobytes, of space used in the dynamic nonpaged pool. SPRMXKBU is the largest amount of space used in the pool.

S P R M X L E I — Maximum Local Event Flag Wait, Inswap

SPRMXLEI contains the maximum count of processes waiting for a local event flag to be signaled.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRLEF - Local Event Flag Wait, Inswapped

ELEMENT DERIVATION: SPRMXLEI is set to the maximum value reached by the MNR_STA$L_LEF field from the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXLEI takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXLEI contains the maximum value of SPRMXLEI processed for the level of summarization.

USAGE NOTES: SPRMXLEI records the maximum count of processes that were waiting for resources that were locally created and managed. Most often, the processes in local event flag wait are waiting for user input at the terminal. Processes included in SPRMXLEI do not affect system performance much.

S P R M X L E O — Maximum Local Event Flag Wait, Outswap

SPRMXLEO contains the maximum count of processes that were waiting for a local event flag to be signaled and were swapped out of memory.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRLEFO - Local Event Flag Wait, Outswapped

ELEMENT DERIVATION: SPRMXLEO is set to the maximum value reached by the MNR_STA$L_LEFO field from the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXLEO takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXLEO contains the maximum value of SPRMXLEO processed for the level of summarization.

USAGE NOTES: SPRMXLEO records the maximum count of processes that were waiting for resources that were locally created and managed. Most often, the processes in local event flag wait are waiting for user input at the terminal. Processes included in SPRMXLEO do not affect system performance much.

For the processes included in SPRMXLEO, when the local event occurs the performance of the process will be affected by the need to swap the process back into memory. This delay, which directly affects the response the user sees, is the only reason why a high count in SPRMXLEO would require attention. If the users are not complaining about poor response time,

this value is of little importance.

| S P R M X L P U | Maximum Large Request Packets in Use

SPRMXLPU contains the maximum number of large request packets (LRPs) in use.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRLPUSE - Large Request Packets in Use

ELEMENT DERIVATION: SPRMXLPU is set to the value of the MNR_POO$L_LRPINUSE field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRMXLPU takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMXLPU contains the maximum value of SPRMXLPU processed for the level of summarization.

USAGE NOTES: The SYSGEN parameter LRPCOUNT determines how many LRPs are preallocated for use by processes on the system. When a process requests pool space that fits into an LRP, one is allocated to the process. SPRLPUSE is the largest number of LRPs in use.

| S P R M X M R W | Maximum Miscellaneous Resource Wait

SPRMXMRW contains the maximum count of processes waiting on resources other than paging lists, event flags, timers, or CPU access.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMWAIT - Miscellaneous Resource Wait

ELEMENT DERIVATION: SPRMXMRW is set to the maximum value reached by the MNR_STA$L_MWAIT field from the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXMRW takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXMRW contains the maximum value of SPRMXMRW processed for the level of summarization.

USAGE NOTES: The processes counted in SPRMXMRW are waiting for resources that ordinarily should not be busy. SPRMXMRW represents processes that are involuntarily forced to wait for system-wide resources such as mailboxes, asynchronous trap mechanisms, or swap and page files. If SPRMXMRW is greater than 4, the VMS system parameters and memory management files should be examined for areas of improvement. The areas to look at first are the size of the page and swap files and the number of I/Os on the disks where those files reside.

| S P R M X O S T | Maximum Other States

SPRMXOST contains the sum of processes in the following states: collided page, common even flag, suspended (inswapped and outswapped), free page wait, and current.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRCOLPG - Collided Page Wait
SPRCEF - Common Event Flag Wait
SPRSUSP - Suspended Wait, Inswapped
SPRSUSPO - Suspended Wait, Outswapped
SPRFPGWT - Free Page Wait
SPRCUR - Current Processes

ELEMENT DERIVATION: SPRMXOST is set to the value of the MNR_SYS$L_OTHSTAT field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE: SPRMXOST takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMXOST contains the maximum value of SPRMXOST processed for the level of summarization.

USAGE NOTES: The states represented here are usually empty, with the exception of current processes. If SPRMXOST is larger than the number of processors on the system, the pattern of SPROSTAT should be observed. When this is the case, there is usually some process that was suspended by the VAX system manager and then forgotten.

| S P R M X P F W | Maximum Page Fault Wait

SPRMXPFW contains the maximum count of processes that were waiting for pages of memory that were on disk.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPFW - Page Fault Wait

ELEMENT DERIVATION: SPRMXPFW is set to the maximum value reached by the MNR_STA$L_PFW field of the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXPFW takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXPFW contains the maximum value of SPRMXPFW processed for the level of summarization.

USAGE NOTES: A high value in SPRMXPFW indicates that the process working set sizes are not set high enough or that many large images are being invoked. If there is still memory on the free page list, the process working sets should be increased to help lower page fault waiting.

If SPRMXPFW is high and there is no memory on the free page list, then the system may need additional physical memory. If SPRMXPFW is high, the free page list is nearly empty and only a few processes are outswapped, the system is probably just invoking many large images.

A high value in SPRMXPFW will have negative effects on the CPU and the I/O subsystem. The CPU usage will not show in any of the processes on the system but will be recorded as kernel and executive mode processing.

| S P R M X P R C | Maximum Process Count

SPRMXPRC contains the maximum number of processes on the system.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPRCNT - Process Count

ELEMENT DERIVATION: SPRMXPRC is set to the value of the MNR_SYS$L_PROCS field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE: SPRMXPRC takes any integer value, 3 to the SYSGEN parameter MAXPROCESSCNT.

SUMMARIZATION PROCESS: SPRMXPRC contains the maximum value of SPRMXPRC processed for the level of summarization.

USAGE NOTES: SPRMXPRC should be equal to the sum of the values in the VMS MONITOR STATES record. The smallest value for SPRMXPRC will vary depending on the number of print SYMBIONTs, network services provided, and diagnostic software in use. If SPRMXPRC is equal to MAXPROCESSCNT and system performance was not a problem, consider increasing MAXPROCESSCNT so that users are not being artificially restricted from the machine.

| S P R M X S M B | Maximum Block Less Than or Equal to 32

SPRMXSMB contains the maximum number of blocks allocated in the dynamic nonpaged pool that were less than or equal to 32 bytes in size.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRSMBLK - Block Less Than or Equal to 32

ELEMENT DERIVATION: SPRMXSMB is set to the value of the MNR_POO$L_SMALLCNT field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRMXSMB takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMXSMB contains the maximum value of SPRMXSMB processed for the level of summarization.

USAGE NOTES: As the dynamic pool gets fragmented, little blocks are formed. Typically, most requests are for more than 32 bytes in size, so the number in SPRMXSMB is the largest number of relatively useless blocks on the dynamic nonpaged pool.

| S P R M X S P I | Maximum Suspended Wait, Inswapped

SPRMXSPI contains the maximum count of processes that were suspended at their own request or at the request of some other process.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRSUSP - Suspended Wait, Inswapped

ELEMENT DERIVATION: SPRMXSPI is set to the maximum value reached by the MNR_STA$L_SUSP field from the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXSPI takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXSPI contains the maximum value of SPRMXSPI processed in the level of summarization.

USAGE NOTES: The count in SPRMXSPI does not affect VMS performance to any great extent. If the count is large and the maximum number of processes allowed on the system is reached, then processes that are suspended take up process slots without doing any work at that time. Suspended processes usually will do useful work when they are taken off suspension, so it is important to allow suspended processes to remain on the system.

| S P R M X S P O | Maximum Suspended Wait, Outswapped

SPRMXSPO contains the maximum count of processes that were suspended at their own request, or at the request of some other process and were then swapped out of memory.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRSUSPO - Suspended Wait, Outswapped

ELEMENT DERIVATION: SPRMXSPO is set to the maximum value reached by the MNR_STA$L_SUSPO field from the VMS MONITOR STATES record during the sampling period.

RANGE OF VALUE: SPRMXSPO takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRMXSPO contains the maximum value of SPRMXSPO processed in the level of summarization.

USAGE NOTES: The count in SPRMXSPO does not affect VMS performance to any great extent. If the count is large and the maximum number of processes allowed on the system is reached, then processes that are suspended take up process slots without doing any work at that time. Suspended processes usually will do useful work when they are taken off suspension, so it is important to allow suspended processes to remain on the system.

Of all the processes to have in an outswapped state, the processes counted in SPRMXSPO are the least costly. Since the process is indefinitely suspended, the extra time of inswapping will have minimal effect on system performance.

| S P R M X S P U | Maximum Small Request Packets in Use

SPRMXSPU contains the maximum number of small request packets (SRPs) in use.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRSPUSE - Small Request Packets in Use

ELEMENT DERIVATION: SPRMXSPU is set to the value of the MNR_POO$L_SRPINUSE field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRMXSPU takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRMXSPU contains the maximum value of SPRMXSPU processed for the level of summarization.

USAGE NOTES: The SYSGEN parameter SRPCOUNT determines how many SRPs are preallocated for use by processes on the system. When a process requests pool space that fits into an SRP, one is allocated to the process. SPRMXSPU is the largest number of SRPs in use.

| S P R N E I | New Enqueue Incoming

SPRNEI contains the count of incoming enqueues that originated on remote nodes and are performed on the recording node.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRNEO - New Enqueue Outgoing
SPRNEL - New Enqueue Local

ELEMENT DERIVATION: SPRNEI is set to the value of the MNR_CLU$L_ENQNEWIN field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRNEI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRNEI contains the sum of all values of SPRNEI processed in the level of summarization.

USAGE NOTES: None

| S P R N E L | New Enqueue Local

SPRNEL contains the count of new local enqueues that originated and are performed on the local node.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRNEO - New Enqueue Outgoing
SPRNEI - New Enqueue Incoming

ELEMENT DERIVATION: SPRNEL is set to the value of the MNR_CLU$L_ENQCNEWLOC field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRNEL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRNEL contains the sum of all values of SPRNEL processed in the level of summarization.

USAGE NOTES: None

| S P R N E N Q | New Enqueues

SPRNENQ contains the number of new enqueues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSNEQ - New Enqueue Rate

ELEMENT DERIVATION: SPRNENQ is set to the value of the MNR_LCK$L_ENQNEW field of the VMS MONITOR LOCK record.

RANGE OF VALUE: SPRNENQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRNENQ contains the sum of all values of SPRNENQ processed in the level of summarization.

USAGE NOTES: When a process requests a resource via an enqueue, SPRNENQ is incremented. If the resource is unavailable, and the enqueue request did not specify 'wait', SPRENQNQ is incremented. If the resource is available to be enqueued, SPRENQWT is incremented. The number of enqueues that got immediate locks can be determined by subtracting SPRENQNQ and SPRENQWT from SPRNENQ.

| S P R N E O | New Enqueue Outgoing

SPRNEO contains the count of new outgoing enqueues that originated on the recording node and were performed on the remote nodes.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRNEI - New Enqueue Incoming
SPRNEL - New Enqueue Local

ELEMENT DERIVATION: SPRNEO is set to the value of the MNR_CLU$L_ENQNEWOUT field of the VMS MONITOR CLUSTER record.

RANGE OF VALUE: SPRNEO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRNEO contains the sum of all values of SPRNEO processed in the level of summarization.

USAGE NOTES: None

```
+-----------------+
| S P R N E W F L |  New Files
+-----------------+
SPRNEWFL contains the number of new files created by the file
system during DURATION (measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSNFL - New Files Rate
           SPRALLOC - Disk Allocations
           SPRACCS  - File Access

ELEMENT DERIVATION:  SPRNEWFL is set to the value of the
MNR_FCP$L_CREATE field of the VMS MONITOR FCP record.

RANGE OF VALUE:  SPRNEWFL takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRNEWFL contains the sum of all
values of SPRNEWFL processed in the level of summarization.

USAGE NOTES:  When a file is created, the file system
determines if there is a requested allocation size; if not,
the default size is used.  The directory is scanned to
determine whether a file of the same name is already present.
If the file exists, the new file is given a version number
one higher than the existing file.  Space is made for the
name of the file as an entry in the directory, space is
allocated, and the new directory entry is pointed at the
storage space.  If the directory is in cache, these
activities require only two physical I/Os:  allocating the
space and writing out the directory on completion.

If SPRNEWFL is high, check to see if the system is spending a
lot of CPU time on the file system.  In addition, a high
value in SPRNEWFL should be a signal to check for disk
fragmentation.  If SPRNEWFL and SPRTURN are both high, the
system may be thrashing on the file map.


+-----------------+
| S P R N L I     |  New Locks, Incoming
+-----------------+
SPRNLI contains the count of incoming enqueues.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRNLO - New Locks, Outgoing
           SPRNLL - New Locks, Local

ELEMENT DERIVATION:  SPRNLI is set to the value of the
MNR_DLO$L_ENQNEWIN field of the VMS MONITOR DLOCK record.

RANGE OF VALUE:  SPRNLI takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRNLI contains the sum of all values
of SPRNLI processed in the level of summarization.

USAGE NOTES:  SPRNLI counts the number of requested locks
(enqueues) received from remote nodes.


+-----------------+
| S P R N L L     |  New Locks, Local
+-----------------+
SPRNLL contains the count of new local enqueues.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRNLO - New Locks, Outgoing
           SPRNLI - New Locks, Incoming

ELEMENT DERIVATION:  SPRNLL is set to the value of the
MNR_DLO$L_ENQCNEWLOC field of the VMS MONITOR DLOCK record.

RANGE OF VALUE:  SPRNLL takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRNLL contains the sum of all values
of SPRNLL processed in the level of summarization.

USAGE NOTES:  SPRNLL counts the number of requested locks
(enqueues) from this node that were newly queued.


+-----------------+
| S P R N L O     |  New Locks, Outgoing
+-----------------+
SPRNLO contains the count of new outgoing enqueues.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRNLI - New Locks, Incoming
```

SPRNLL - New Locks, Local

ELEMENT DERIVATION: SPRNLO is set to the value of the MNR_DLO$L_ENQNEWOUT field of the VMS MONITOR DLOCK record.

RANGE OF VALUE: SPRNLO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRNLO contains the sum of all values of SPRNLO processed in the level of summarization.

USAGE NOTES: SPRNLO counts the number of requested locks (enqueues) newly sent to remote nodes.

S P R O P E N | Files Opened

SPROPEN contains the number of files opened during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSOPN - File Open Rate
SPRFILOP - Files Opened

ELEMENT DERIVATION: SPROPEN is set to the value of the MNR_FCP$L_OPENS field of the VMS MONITOR FCP record.

RANGE OF VALUE: SPROPEN takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPROPEN contains the sum of all values of SPROPEN processed in the level of summarization.

USAGE NOTES: Opening files can consume a lot of resources, both in CPU time and the number of disk I/Os required. If the directory of the file is cached in memory, the resource requirement is less. When a file is opened, the location is looked up in the directory. If the directory is not in cache, the directory must first be read from disk into cache. Once the location of the file is found, the file header is then read into the file header cache, and the file's table of block locations and index buffers are allocated and read from disk.

If SPROPEN is large, a lot of the machine resources are being used to initialize the use of files. If this seems to be affecting performance, the VAX system manager can do one of two things: increase directory caches, which takes away user memory, or modify applications so they do not open as many files.

S P R O S T A T | Other States

SPROSTAT contains the sum of processes in the following states: collided page, common even flag, suspended (inswapped and outswapped), free page wait, and current.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRCOLPG - Collided Page Wait
SPRCEF - Common Event Flag Wait
SPRSUSP - Suspended Wait, Inswapped
SPRSUSPO - Suspended Wait, Outswapped
SPRFPGWT - Free Page Wait
SPRCUR - Current Processes

ELEMENT DERIVATION: SPROSTAT is set to the value of the MNR_SYS$L_OTHSTAT field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE: SPROSTAT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPROSTAT contains the last value of SPROSTAT processed in the level of summarization.

USAGE NOTES: The states represented here are usually empty, with the exception of current processes. If SPROSTAT is consistently larger than the number of processors on the system, there is something that should be investigated. When this is the case, there is usually some process that was suspended by the VAX system manager and then forgotten.

S P R P C B I T | Storage Bitmap Cache Hit Percentage

SPRPCBIT contains the storage bitmap cache hit percentage.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRBITHT - Storage Bitmap Cache Hits

ELEMENT DERIVATION: SPRPCBIT is computed using the following

algorithm:

```
  if SPRBITAT > 0 then
    SPRPCBIT = SPRBITHT * 100 / SPRBITAT
  else
    SPRPCBIT = 0
```

RANGE OF VALUE: The value of SPRPCBIT ranges from 0 to 100.

SUMMARIZATION PROCESS: SPRPCBIT contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When disk space is allocated for a file, the first place searched is the extent cache. If there is not enough space there, the storage bitmap is tried. The storage bitmap cache is searched for an area large enough to hold the file. This search occurrence is recorded in SPRBITAT. If the space is found in the cache, the occurrence is recorded in SPRBITHT. Activity on the storage bitmap is an indication that the disk volume is fragmented. The objective is to have most of the disk space allocation satisfied in extent cache. If SPRBITAT is much larger than SPRBITHT, the disk is very badly fragmented and should be defragmented. The size of the bitmap cache is set by the SYSGEN parameter ACP_MAPCACHE. The closer SPRPCBIT is to 0 and the closer SPRPCEXT is to 100, the better the performance is for this operation.

S P R P C D D C — Directory Data Cache Hit Percentage

SPRPCDDC contains the directory data cache hit percentage.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRDATHT - Directory Data Cache Hits

ELEMENT DERIVATION: SPRPCDDC is computed using the following algorithm:

```
  if SPRDATAT > 0 then
    SPRPCDDC = SPRDATHT * 100 / SPRDATAT
  else
    SPRPCDDC = 0
```

RANGE OF VALUE: The value of SPRPCDDC ranges from 0 to 100.

SUMMARIZATION PROCESS: SPRPCDDC contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When a file is opened, created, deleted, extended, or truncated, the file name is looked up in the directory. The directory cache is searched for the directory data needed. This occurrence is recorded in SPRDATAT. If the directory is found in the cache, the occurrence is recorded in SPRDATHT. The directory data cache size is set by the SYSGEN parameter ACP_DIRCACHE. The closer SPRPCDDC is to 100, the better the performance is for this operation.

S P R P C E X T — Extent Cache Hit Percentage

SPRPCEXT contains the extent cache hit percentage.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPREXTHT - Extent Cache Hits

ELEMENT DERIVATION: SPRPCEXT is computed using the following algorithm:

```
  if SPREXTAT > 0 then
    SPRPCEXT = SPREXTHT * 100 / SPREXTAT
  else
    SPRPCEXT = 0
```

RANGE OF VALUE: The value of SPRPCEXT ranges from 0 to 100.

SUMMARIZATION PROCESS: SPRPCEXT contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When disk space is allocated for a file, the first place searched is the extent cache. The occurrence is recorded in SPREXTAT. If there is a large enough space to satisfy the allocation, the space is allocated in extent cache and the occurrence is recorded in SPREXTHT. When a file is deleted or truncated, an attempt is made to put the space back into the extent cache. This occurrence is recorded in SPREXTAT. If the extent cache can hold the space being returned, the occurrence is recorded in SPREXTHT. If SPREXTAT is much larger than SPREXTHT, either the disk is fragmented or the extent cache is too small. The size of the extent cache is set by the SYSGEN parameters ACP_EXTCACHE and ACP_EXTLIMIT. The closer SPRPCEXT is to 100, the better the performance is for this operation.

```
+-----------------+
| S P R P C F C B |  Directory FCB Cache Hit Percentage
+-----------------+
SPRPCFCB contains the directory file control block (FCB)
cache hit percentage.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRFCBHT - Directory FCB Cache Hits

ELEMENT DERIVATION:  SPRPCFCB is computed using the following
algorithm:

  if SPRFCBAT > 0 then
    SPRPCFCB = SPRFCBHT * 100 / SPRFCBAT
  else
    SPRPCFCB = 0

RANGE OF VALUE:  The value of SPRPCFCB ranges from 0 to 100.

SUMMARIZATION PROCESS:  SPRPCFCB contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  When a file is opened, created, deleted,
extended, or truncated, the system has to find the directory
containing the file name.  The directory FCB cache is
searched for the FCB describing the file.  This occurrence is
recorded in SPRFCBAT.  If the FCB is found, the occurrence is
recorded in SPRFCBHT.  The size of the directory FCB cache is
set by the SYSGEN parameters ACP_SYSACC and ACP_DINDXCACHE.
The closer SPRPCFCB is to 100, the better the performance is
for this operation.


+-----------------+
| S P R P C F H C |  File Header Cache Hit Percentage
+-----------------+
SPRPCFHC contains the file header cache hit percentage.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRFHDHT - File Header Cache Hits

ELEMENT DERIVATION:  SPRPCFHC is computed using the following
algorithm:

  if SPRFHDAT > 0 then
    SPRPCFHC = SPRFHDHT * 100 / SPRFHDAT
  else
    SPRPCFHC = 0

RANGE OF VALUE:  The value of SPRPCFHC ranges from 0 to 100.

SUMMARIZATION PROCESS:  SPRPCFHC contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  When an open, close, extend, delete, or
truncate operation is requested on a file, the file header is
located by searching the file header cache.  The search
occurrence is recorded in SPRFHDAT.  If the header is found,
the occurrence is recorded in SPRFHDHT.  The size of the file
header cache is set by the SYSGEN parameter ACP_HDRCACHE.
The closer SPRPCFHC is to 100, the better the performance is
for this operation.


+-----------------+
| S P R P C F I D |  File ID Cache Hit Percentage
+-----------------+
SPRPCFID contains the file ID cache hit percentage.

TYPE: Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO: SPRFIDHT - File ID Cache Hits

ELEMENT DERIVATION: SPRPCFID is computed using the following
algorithm:

  if SPRFIDAT > 0 then
    SPRPCFID = SPRFIDHT * 100 / SPRFIDAT
  else
    SPRPCFID = 0

RANGE OF VALUE: The value of SPRPCFID ranges from 0 to 100.

SUMMARIZATION PROCESS: SPRPCFID contains the result of the
computation described above in ELEMENT DERIVATION. The value
is recomputed at each level of summarization.

USAGE NOTES: File ID numbers are issued when files are
created and returned on deletion.  When a file is created,
the file ID cache is searched for the next file ID.  The
occurrence is recorded in SPRFIDAT.  If the file ID is there,
the occurrence is recorded in SPRFIDHT.  When a file is
deleted, its file ID is returned to the file ID cache.  The
```

return occurrence is recorded in SPRFIDAT. If the file ID can be returned to the cache, the occurrence is recorded in SPRFIDHT. If SPRFIDAT is much larger than SPRFIDHT, the performance of creating and deleting files is hampered. The size of the file ID cache is set by the SYSGEN parameter ACP_FIDCACHE. The closer SPRPCFID is to 100, the better the performance is for this operation.

S P R P C Q O T — Quota Cache Hit Percentage

SPRPCQOT contains the quota cache hit percentage.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRQUOHT - Quota Cache Hits

ELEMENT DERIVATION: SPRPCQOT is computed using the following algorithm:

```
  if SPRQUOAT > 0 then
    SPRPCQOT = SPRQUOHT * 100 / SPRQUOAT
  else
    SPRPCQOT = 0
```

RANGE OF VALUE: The value of SPRPCQOT ranges from 0 to 100.

SUMMARIZATION PROCESS: SPRPCQOT contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: If quota checking is enabled, a change in file size causes a search of the quota cache for an entry describing the file owner. The search occurrence is recorded in SPRQUOAT. If the entry is found, the occurrence is recorded in SPRQUOHT. If SPRQUOAT is much larger than SPRQUOHT, then quota cache is not large enough and quota checking could be causing performance problems. The size of the quota cache is set by the SYSGEN ACP_QUOCACHE. The closer SPRPCQOT is to 100, the better the performance is for this operation.

S P R P F W — Page Fault Wait

SPRPFW contains the count of processes waiting for pages of memory that are currently on disk.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRCOLPG - Collided Page Wait
SPRFPGWT - Free Page Wait
SPRMXPFW - Maximum Page Fault Wait

ELEMENT DERIVATION: SPRPFW is set to the value of the MNR_STA$L_PFW field of the VMS MONITOR STATES record.

RANGE OF VALUE: SPRPFW takes any integer value from 0 to the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS: SPRPFW contains the last value of SPRPFW processed in the level of summarization.

USAGE NOTES: A high value in SPRPFW indicates that the process working set sizes are not set high enough or that many large images are being invoked. If there is still memory on the free page list, the process working sets should be increased to help lower page fault waiting.

If SPRPFW is high and there is no memory on the free page list, then the system may need additional physical memory. If SPRPFW is high, the free page list is not empty and only a few processes are outswapped, the system is probably just invoking many large images.

A high value in SPRPFW will have negative effects on the CPU and the I/O subsystem. The CPU usage will not show in any of the processes on the system but will be recorded as kernel and executive mode processing.

S P R P G E C T — PGE PAGE Record Count (Type 3)

SPRPGECT contains a count of VMS MONITOR PAGE records processed by the CA MICS VAX/VMS Analyzer.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS MONITOR PAGE record is

stored, SPRPGECT is set to 1.

RANGE OF VALUE: In the DETAIL timespan of the CA MICS VMS System Profile file, SPRPGECT is always 1. At higher levels of summarization, its value ranges from 1 to the total number of VMS MONITOR PAGE records processed in the level of summarization

SUMMARIZATION PROCESS: SPRPGECT contains the sum of all values of SPRPGECT processed in the level of summarization.

USAGE NOTES: SPRPGECT is used as the divisor for calculating averages and ratios in the level of summarization.

S P R P G E D R | PGE PAGE Record Duration (Type 3)

SPRPGEDR contains the number of seconds in the "reporting interval" of a given VMS MONITOR record. Reporting interval is the period of time during which a monitor gathers and reports statistics on this record.

TYPE: Common accumulated data element - Numeric, measurement unit is integer.

SEE ALSO:
- SPRPGECT - PGE PAGE Record Count (Type 3)
- INTERVLS - Number of Recording Intervals
- STARTTS - Start Time Stamp
- ENDTS - End Time Stamp
- DEXDUR - Requested Recording Interval Time

ELEMENT DERIVATION: SPRPGEDR contains the difference between the time stamp of the current VMS MONITOR record and the time stamp of the previously processed record of the same type.

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SPRPGEDR contains the sum of all occurrences of SPRPGEDR that were included in the level of summarization.

USAGE NOTES: Note that SPRPGEDR is uniquely calculated for this VMS MONITOR record type before merging records together into the DEMSPR file.

S P R P G E F L | PGE Page Faults

SPRPGEFL contains the number of pages allocated to processes as a result of page faults during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO:
- SPRIOVFL - IO Page Faults
- SPRFCPFL - FCP Page Faults
- SPRPGFLT - Page Faults
- PRCPAGFT - Page Faults
- IMGPAGFT - Page Faults
- SPRPSPFL - PGE Page Fault Rate

ELEMENT DERIVATION: SPRPGEFL is set to the value of the MNR_PAG$L_FAULTS field of the VMS MONITOR PAGE record.

RANGE OF VALUE: SPRPGEFL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPGEFL contains the sum of all values of SPRPGEFL processed in the level of summarization.

USAGE NOTES: SPRPGEFL includes all page faults on the system, including the other faults recorded in SPRFPGFL, SPRMPGFL, SPRDMZFL, SPRGBLFL, and SPRWIPFL. High page faulting can cause negative effects on the CPU and I/O subsystem in VMS. To reduce page faulting, the VMS system manager should try to adjust the user authorization file parameters WSDEFAULT, WSQUOTA, and WSEXTENT and the SYSGEN parameters PFRATH, BORROWLIM, and WSINC.

WSDEFAULT, WSQUOTA, and WSEXTENT describe the number of physical pages allocated to a process. WSDEFAULT is the amount with which the process starts. WSQUOTA is the amount of memory allowed for the process, regardless of the memory usage on the system. WSEXTENT is the amount of memory the process may borrow when there is free memory to spare.

PFRATH is the number of page faults per 10-second interval that a process must exceed to be eligible to borrow free memory. BORROWLIM is the amount of free memory that must be available before any will be loaned. WSINC is the number of memory pages that are loaned with each increase.

```
+-----------------+
| S P R P G E R D |  PGE Page Reads
+-----------------+
SPRPGERD contains the number of pages read from disk to
satisfy page faults during DURATION (measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPGERI - PGE Page Read I/Os
           SPRPGRD  - Page Reads
           SPRPSRED - PGE Page Read Rate

ELEMENT DERIVATION:  SPRPGERD is set to the value of the
MNR_PAG$L_PREADS field of the VMS MONITOR PAGE record.

RANGE OF VALUE:  SPRPGERD takes any integer value from 0 to
SPRPGEFL.

SUMMARIZATION PROCESS:  SPRPGERD contains the sum of all
values of SPRPGERD processed in the level of summarization.

USAGE NOTES:  SPRPGERD is a count of pages read from disk,
not the number of I/O transactions.  VMS will try to package
several page fault reads into one I/O transaction if
possible.  Nevertheless, SPRPGERD are much more harmful to
performance than the faults that can be satisfied in memory,
such as SPRFPGFL, SPRMPGFL or SPRGBLFL.  For every page in
SPRPGERD, there was also a process counted in STAFLT;
however, the converse is not true.


+-----------------+
| S P R P G E R I |  PGE Page Read I/Os
+-----------------+
SPRPGERI contains the number of read I/O operations from disk
to satisfy page faults during DURATION (measurement
interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPGERD - PGE Page Reads
           SPRPRDIO - Page Read I/Os
           SPRFCPRI - FCP Disk Read I/Os
           SPRPSPRI - PGE Page Read I/O Rate

ELEMENT DERIVATION:  SPRPGERI is set to the value of the
MNR_PAG$L_PREADIO field of the VMS MONITOR PAGE record.

RANGE OF VALUE:  SPRPGERI takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPGERI contains the sum of all
values of SPRPGERI processed in the level of summarization.

USAGE NOTES:  When a page fault causes a page to be read from
disk, the memory manager determines if any other reads can be
combined with it to reduce the number of actual I/O
transactions.  Thus, SPRPGERI can never exceed SPRPGERD.  If
SPRPGEFL is high and SPRPGERI is nearly equal to SPRPGERD,
then the page reading is particularly inefficient.  To
improve efficiency, it may be necessary to adjust some file
caching parameters.


+-----------------+
| S P R P G E W I |  PGE Page Write I/Os
+-----------------+
SPRPGEWI contains the number of write operations to the page
file to make room for requested pages.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSPWI - PGE Page Write I/O Rate
           SPRPGEWR - PGE Page Writes
           SPRPWIO  - Page Write I/Os
           SPRFCPWI - FCP Disk Write I/Os

ELEMENT DERIVATION:  SPRPGEWI is set to the value of the
MNR_PAG$L_PWRITIO field of the VMS MONITOR PAGE record.

RANGE OF VALUE:  SPRPGEWI takes any integer value from 0 to
the value of SPRPGEWR.

SUMMARIZATION PROCESS:  SPRPGEWI contains the sum of all
values of SPRPGEWI processed in the level of summarization.

USAGE NOTES:  When the memory manager needs to write a page
to the page file, it checks to see if there are other pages
it can write in the same operation.  Therefore, SPRPGEWI can
never be larger than SPRPGEWR.


+-----------------+
| S P R P G E W R |  PGE Page Writes
+-----------------+
SPRPGEWR contains the number of pages written to the page
file in DURATION (measurement interval).
```

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSWRT - PGE Page Write Rate
SPRPGEWI - PGE Page Write I/Os
SPRPGWT - Page Writes

ELEMENT DERIVATION: SPRPGEWR is set to the value of the MNR_PAG$L_PWRITES field of the VMS MONITOR PAGE record.

RANGE OF VALUE: SPRPGEWR takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPGEWR contains the sum of all values of SPRPGEWR processed in the level of summarization.

USAGE NOTES: SPRPGEWR is a sign of hard page faulting because memory is being written out to disk to make room for other memory to be allocated. If the data in the pages is needed again, then those pages will have to be read in once more.

S P R P G F L T | Page Faults

SPRPGFLT contains the count of pages allocated to processes as a result of page faults during DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSFLT - Page Fault Rate
SPRPGEFL - PGE Page Faults
SPRFCPFL - FCP Page Faults
SPRIOVFL - IO Page Faults
PRCPAGFT - Page Faults
IMGPAGFT - Page Faults

ELEMENT DERIVATION: SPRPGFLT is set to the value of the MNR_SYS$L_FAULTS field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE: SPRPGFLT takes an integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPGFLT contains the sum of all values of SPRPGFLT processed in the level of summarization.

USAGE NOTES: High page faulting can cause negative effects on the CPU and I/O subsystem of VAX/VMS. To reduce page faulting, the VAX system manager should try to adjust the user authorization file parameters WSDEFAULT, WSQUOTA, and WSEXTENT, along with the SYSGEN parameters PFRATH, BORROWLIM, and WSINC.

WSDEFAULT, WSQUOTA, and WSEXTENT describe the number of physical pages allocated to a process. WSDEFAULT is the amount with which the process starts. WSQUOTA is the amount of memory allowed for the process, regardless of the memory usage on the system. WSEXTENT is the limit to the amount of memory the process may borrow when there is memory to spare.

PFRATH is the number of page faults per 10-second interval that a process must exceed to be eligible to borrow free memory. BORROWLIM is the amount of free memory that must be available before any will be loaned. WSINC is the number of memory pages that are loaned with each increase.

S P R P G R D | Page Reads

SPRPGRD contains the count of pages read from disk to satisfy page faults for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPRDIO - Page Read I/Os
SPRPSPGR - Page Read Rate
SPRPGERD - PGE Page Reads

ELEMENT DERIVATION: SPRPGRD is set to the value of the MNR_IO$L_PREADS field of the VMS MONITOR IO record.

RANGE OF VALUE: SPRPGRD takes an integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPGRD contains the sum of all values of SPRPGRD processed in the level of summarization.

USAGE NOTES: SPRPGRD is a count of pages read from disk not the number of I/O transactions. VMS will try to package several page fault reads into one I/O transaction if possible. Nevertheless, SPRPGRD is much more harmful to performance than the faults that can be satisfied in memory, such as SPRFPGFL, SPRMPGFL, or SPRGBLFL. For every page in SPRPGRD, there was also a process counted in STAFLT; however, the converse is not true.

| S P R P G W T | Page Writes

SPRPGWT contains the count of pages written to the page file in DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSPGW - Page Write Rate
SPRPWIO - Page Write I/Os
SPRPGEWR - PGE Page Writes

ELEMENT DERIVATION: SPRPGWT is set to the value of the MNR_IO$L_PWRITES field of the VMS MONITOR IO record.

RANGE OF VALUE: SPRPGWT takes an integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPGWT contains the sum of all values of SPRPGWT processed in the level of summarization.

USAGE NOTES: SPRPGWT is an indication of heavy page faulting because memory is being written out to disk to make room for other memory to be allocated. If the data in the pages is needed again, those pages will have to be read in once more.

| S P R P K L O S | Transit Packets Lost

SPRPKLOS contains the number of transit packets lost.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSLOS - Transit Packet Loss Rate

ELEMENT DERIVATION: SPRPKLOS is set to the value of the MNR_NET$L_TRCNGLOS field of the VMS MONITOR DECNET record.

RANGE OF VALUE: SPRPKLOS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPKLOS contains the sum of all values of SPRPKLOS processed in the level of summarization.

USAGE NOTES: When the network has a high volume of traffic, packets may be lost during transmission. DECnet provides the capability to recognize when a packet sent from one node is not received by another node. When this occurs, the packet is retransmitted to the original node destination. SPRPKLOS contains the number of packets retransmitted because the network could not process all the incoming packets.

| S P R P O L C T | POL POOL Record Count (Type 6)

SPRPOLCT contains a count of pool records processed by the CA MICS VAX/VMS Analyzer.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS MONITOR POOL record is processed, SPRPOLCT is set to 1.

RANGE OF VALUE: In the DETAIL timespan of the CA MICS VMS System Profile file, SPRPOLCT is always 0 or 1. At higher levels of summarization, its value ranges from 0 to the total number of pool records processed in the level of summarization.

SUMMARIZATION PROCESS: SPRPOLCT contains the sum of all values of SPRPOLCT processed in the level of summarization.

USAGE NOTES: SPRPOLCT is used as a divisor for calculating averages and ratios in the level of detail summarization.

| S P R P O L D R | POL POOL Record Duration (Type 6)

SPRPOLDR contains the number of seconds in the "reporting interval" of a given VMS MONITOR record. Reporting interval is the period of time during which a monitor gathers and reports statistics on this record.

TYPE: Common accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: SPRPOLCT - POL POOL Record Count (Type 6)
INTERVLS - Number of Recording Intervals
STARTTS - Start Time Stamp
ENDTS - End Time Stamp
DEXDUR - Requested Recording Interval Time

ELEMENT DERIVATION: SPRPOLDR contains the difference between the time stamp of the current VMS MONITOR record and the time stamp of the previously processed record of the same type.

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SPRPOLDR contains the sum of all occurrences of SPRPOLDR that were included in the level of summarization.

USAGE NOTES: Note that SPRPOLDR is uniquely calculated for this VMS MONITOR record type before merging records together into the DEMSPR file.

S P R P R C N T — Process Count

SPRPRCNT contains the number of processes on the system.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMXPRC - Maximum Process Count

ELEMENT DERIVATION: SPRPRCNT is set to the value of the MNR_SYS$L_PROCS field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE: SPRPRCNT takes any integer value from 3 to the SYSGEN parameter MAXPROCESSCNT.

SUMMARIZATION PROCESS: SPRPRCNT contains the last value of SPRPRCNT processed in the level of summarization.

USAGE NOTES: SPRPRCNT should be equal to the sum of the values in the VMS MONITOR STATES record. The smallest value for SPRPRCNT will vary depending on the number of print SYMBIONTs, network services provided, and diagnostic software in use. If SPRPRCNT is equal to MAXPROCESSCNT and system performance was not a problem, increasing MAXPROCESSCNT should be considered so that users are not being artificially restricted from the machine.

S P R P R D I O — Page Read I/Os

SPRPRDIO contains the count of read I/O operations from disk to satisfy page faults for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPGRD - Page Reads
SPRPSPR - Page Read I/O Rate
SPRFCPRI - FCP Disk Read I/Os
SPRPGERI - PGE Page Read I/Os

ELEMENT DERIVATION: SPRPRDIO is set to the value of the MNR_IO$L_PREADIO field of the VMS MONITOR IO record.

RANGE OF VALUE: SPRPRDIO takes an integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPRDIO contains the sum of all values of SPRPRDIO processed in the level of summarization.

USAGE NOTES: When a page fault causes a page to be read from disk, the memory manager looks to see if any other reads can be combined with it to reduce the number of actual I/O transactions. Thus, SPRPRDIO can never exceed SPRPGRD. If SPRIOVFL is high and SPRPRDIO is nearly equal to SPRPGRD, the page reading is particularly inefficient. To improve efficiency it may be necessary to adjust some file caching parameters.

S P R P S A C C — File Access Rate

SPRPSACC contains the number of directory accesses per second during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRACCS - File Access
SPRPSNFL - New Files Rate
SPRPSTRN - Window Turn Rate
SPRPSOPN - File Open Rate
SPRPSEIO - Erase I/O Rate

ELEMENT DERIVATION: SPRPSACC is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSACC = SPRACCS / DURATION
  else
    SPRPSACC = .
```

RANGE OF VALUE: SPRPSACC takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSACC contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: File accesses are directory lookups for the location of the file. SPRPSACC increases with the rate of files created, extended, opened, deleted, or erased and with window turns. Directory lookups are only costly when the file system cache is not large enough for the directories regularly required on the system. If the SPRPSACC is large, the SYSGEN parameters ACP_SYSACC, ACP_DINDXCACHE, ACP_DIRCACHE, and ACP_MAPCACHE may need adjusting to increase performance.

S P R P S A L L — Disk Allocation Rate

SPRPSALL contains the number of disk allocations per second during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRALLOC - Disk Allocations
SPRPSNFL - New Files Rate
SPRPSFWI - FCP Disk Write I/O Rate

ELEMENT DERIVATION: SPRPSALL is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSALL = SPRALLOC / DURATION
  else
    SPRPSALL = .
```

RANGE OF VALUE: SPRPSALL takes any integer value 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSALL contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: Space is allocated when a file is created or when a write request causes the file to be extended. If SPRPSALL is high, the file system should checked for fragmentation. Fragmentation can severely affect disk I/O performance as well as overall system performance. Disk I/O performance is affected by the extra seek time moving from one portion of a file to another. System performance can be degraded if the fragmentation causes a high rate of window turns.

S P R P S A R L — Arriving Local Packet Rate

SPRPSARL contains the number of arriving local packets per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRARLPK - Arriving Local Packets

ELEMENT DERIVATION: SPRPSARL is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSARL = DNTARL / DURATION
  else
    SPRPSARL = 0
```

RANGE OF VALUE: SPRPSARL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSARL contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRARLPK counts the packets addressed to and received by the recording node. The packets are not just limited to data packets; they also include sequencing and other control packets. SPRPSARL is the rate at which packets arrive at the recording node.

```
+-----------------+
| S P R P S A S T |  Blocking ASTs Rate
+-----------------+
SPRPSAST contains the number of blocking asynchronous system
traps (ASTs) per second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRBLAST - Blocking ASTs

ELEMENT DERIVATION:  SPRPSAST is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSAST = SPRBLAST / DURATION
  else
    SPRPSAST = 0

RANGE OF VALUE:  SPRPSAST takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSAST contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRBLAST is the number of requests for
ASTs to notify local processes that their requests are being
served.  SPRPSAST is the number of times per second that
those requests are entered.


+-----------------+
| S P R P S A T P |  Arriving Transit Packet Rate
+-----------------+
SPRPSATP contains the number of arriving transit packets per
second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRARTPK - Arriving Transit Packets

ELEMENT DERIVATION:  SPRPSATP is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSATP = SPRARTPK / DURATION
  else
    SPRPSATP = 0

RANGE OF VALUE:  SPRPSATP takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSATP contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  Transit packets are those that arrive at this
node to be routed to a subsequent node.  If this node is a
routing node, the ratio between SPRARTPK and SPRARLPK gives a
good estimate of the effect of the routing function on the
node.  If SPRPSATP is high, routing may be causing a load on
the system.


+-----------------+
| S P R P S B A I |  Blocking ASTs Incoming Rate
+-----------------+
SPRPSBAI contains the number of incoming blocking ASTs per
second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRBAI - Blocking ASTs, Incoming

ELEMENT DERIVATION:  SPRPSBAI is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSBAI = SPRBAI / DURATION
  else
    SPRPSBAI = 0

RANGE OF VALUE:  SPRPSBAI takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSBAI contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSBAI is the number of requests per second
for asynchronous system traps to notify processes on remote
nodes that their requests are being served.
```

```
+-----------------+
| S P R P S B A L |  Blocking ASTs Local Rate
+-----------------+
SPRPSBAL contains the number of local blocking ASTs per
second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRBAL - Blocking ASTs, Local

ELEMENT DERIVATION:  SPRPSBAL is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSBAL = SPRBAL / DURATION
  else
    SPRPSBAL = 0

RANGE OF VALUE:  SPRPSBAL takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSBAL contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRBAL is the number of requests per second for
asynchronous system traps to notify local processes that
their requests are being served.


+-----------------+
| S P R P S B A O |  Blocking ASTs Outgoing Rate
+-----------------+
SPRPSBAO contains the number of outgoing blocking ASTs per
second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRBAO - Blocking ASTs, Outgoing

ELEMENT DERIVATION:  SPRPSBAO is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSBAO = SPRBAO / DURATION
  else
    SPRPSBAO = 0

RANGE OF VALUE:  SPRPSBAO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSBAO contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRBAO is the number of requests per second for
asynchronous system traps on remote nodes to notify processes
on this node that their requests are being served.


+-----------------+
| S P R P S B I O |  Buffered I/O Rate
+-----------------+
SPRPSBIO contains the number of buffered I/Os per second for
DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRBIOS - Buffered I/Os

ELEMENT DERIVATION:  SPRPSBIO is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSBIO = SPRBIOS / DURATION
  else
    SPRPSBIO = .

RANGE OF VALUE:  SPRPSBIO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSBIO contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSBIO reflects the number of buffered I/O
operations per second, not the size of those operations.
Buffered I/O operations usually do not affect system
performance much.  Systems with all terminals, printers, and
other buffered devices under direct control of the CPU,
however, can be affected by large numbers of buffered I/Os.
If buffered I/O is a problem on the system, reconfiguring
hardware to include terminal servers will help reduce the
effect on the system.
```

```
+-----------------+
| S P R P S B I T |  Storage Bitmap Cache Attempt Rate
+-----------------+
SPRPSBIT contains the number of times per second the system
made a storage bitmap cache attempt.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRBITAT - Storage Bitmap Cache Attempts

ELEMENT DERIVATION:  SPRPSBIT is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSBIT = SPRBITAT / DURATION
  else
    SPRPSBIT = 0

RANGE OF VALUE:  SPRPSBIT takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSBIT contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  When space is allocated for a file, the first
place searched is the extent cache.  If there is not enough
space there, the storage bitmap is tried.  The storage bitmap
cache is checked for an area large enough for the file.  This
check is registered in SPRBITAT.  SPRPSBIT is the number of
these checks per second.  A value greater than 2 on a regular
basis is a clue that the disk may be badly fragmented.


+-----------------+
| S P R P S B U F |  Buffered I/O Rate
+-----------------+
SPRPSBUF contains the number of buffered I/Os per second for
DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRBUFIO - Buffered I/Os

ELEMENT DERIVATION:  SPRPSBUF is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSBUF = SPRBUFIO / DURATION
  else
    SPRPSBUF = .

RANGE OF VALUE:  SPRPSBUF takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSBUF contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSBUF reflects the number of buffered I/O
operations per second, not the size of those operations.
Buffered I/O operations usually do not affect system
performance much.  Systems with all terminals, printers, and
other buffered devices under direct control of the CPU,
however, can be affected by large numbers of buffered I/Os.
If buffered I/O is a problem on the system, reconfiguring
hardware to include terminal servers will help reduce the
effect to the system.


+-----------------+
| S P R P S C A L |  FCP Call Rate
+-----------------+
SPRPSCAL contains the number of FCP calls per second during
DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRCALLS - FCP Calls

ELEMENT DERIVATION:  SPRPSCAL is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSCAL = SPRCALLS / DURATION else
    SPRPSCAL = .

RANGE OF VALUE:  SPRPSCAL takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSCAL contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRCALLS is an indicator of the activity on the
file system that is due to typical application programs.
```

```
+-----------------+
| S P R P S C E Q |  Converted Enqueue Rate
+-----------------+
SPRPSCEQ contains the number of converted enqueues per
second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRCENQ - Converted Enqueues

ELEMENT DERIVATION:  SPRPSCEQ is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSCEQ = SPRCENQ / DURATION
  else
    SPRPSCEQ = 0

RANGE OF VALUE:  SPRPSCEQ takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSCEQ contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRCENQ counts the number of requested enqueues
that were granted after the process had been waiting.
SPRPSCEQ is the number of times per second that waiting-on-
enqueues were converted to locks.


+-----------------+
| S P R P S D D C |  Directory Data Cache Attempt Rate
+-----------------+
SPRPSDDC contains the number of times per second the system
attempted to locate a file name in the directory data cache.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRDATAT - Directory Data Cache Attempts

ELEMENT DERIVATION:  SPRPSDDC is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSDDC = SPRDATAT / DURATION
  else
    SPRPSDDC = 0


RANGE OF VALUE:  SPRPSDDC takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSDDC contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  When a file is opened, created, deleted,
extended, or truncated, the file name is looked up in the
directory.  The directory cache is searched for the directory
data needed.  The search occurrence is recorded in SPRDATAT.
A large value in SPRPSDDC means that the ACP_DIRCACHE is much
too small or that there is an inordinate amount of file
activity on the system.  Look into both possibilities.


+-----------------+
| S P R P S D E Q |  Dequeue Rate
+-----------------+
SPRPSDEQ contains the number of dequeues per second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRDEQ - Dequeues

ELEMENT DERIVATION:  SPRPSDEQ is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSDEQ = SPRDEQ / DURATION
  else
    SPRPSDEQ = 0

RANGE OF VALUE:  SPRPSDEQ takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSDEQ contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRDEQ counts the number of locks that were
released (dequeued).  SPRPSDEQ is the number of times per
second that resources were released.
```

```
+-----------------+
| S P R P S D F D |  Deadlocks Found Rate
+-----------------+
SPRPSDFD contains the number of deadlocks the Lock Manager
found per second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRDEADF - Deadlocks Found

ELEMENT DERIVATION:  SPRPSDFD is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSDFD = SPRDEADF / DURATION
  else
    SPRPSDFD = 0

RANGE OF VALUE:  SPRPSDFD takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSDFD contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  Deadlocks are a potential problem any time
processes must serialize resources to accomplish their tasks.
Whenever a process holds a resource beyond the time of the
system deadlock wait, the Lock Manager searches all processes
waiting on resources to determine if a deadlock condition
exists.  SPRDEADF is the number of times the Lock Manager
determined that a deadlock existed and it took measures to
clear the blocked resource(s).  SPRPSDFD is the number of
times per second that the Lock Manager discovered deadlocks.


+-----------------+
| S P R P S D I O |  Direct I/O Rate
+-----------------+
SPRPSDIO contains the number of direct I/Os per second for
DURATION (measurement interval).

TYPE: Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO: SPRDIOS - Direct I/Os

ELEMENT DERIVATION: SPRPSDIO is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSDIO = SPRDIOS / DURATION
  else
    SPRPSDIO = .

RANGE OF VALUE: SPRPSDIO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: SPRPSDIO contains the result of the
computation described above in ELEMENT DERIVATION. The value
is recomputed at each level of summarization.

USAGE NOTES: SPRPSDIO reflects the number of direct I/O
operations per second, not the size of those operations.
Direct I/Os can have great effect on system performance, but
the effect varies widely depending on average transaction
size.  If the number of direct I/Os is constant, but average
transaction size is larger, the negative effect will be
greater.


+-----------------+
| S P R P S D I R |  Direct I/O Rate
+-----------------+
SPRPSDIR contains the number of direct I/Os per second for
DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRDIRIO - Direct I/Os

ELEMENT DERIVATION:  SPRPSDIR is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSDIR = SPRDIRIO / DURATION
  else
    SPRPSDIR = .

RANGE OF VALUE:  SPRPSDIR takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSDIR contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSDIR reflects the number of direct I/O
operations per second not the size of those operations.
Direct I/Os can have great effect on system performance, but
```

the effect varies widely depending on average transaction size. If the number of direct I/Os is constant, but average transaction size is larger, the negative effect will be greater.

S P R P S D L M | Deadlock Message Rate

SPRPSDLM contains the number of deadlock messages per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRDLM - Deadlock Messages

ELEMENT DERIVATION: SPRPSDLM is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSDLM = SPRDLM / DURATION
  else
    SPRPSDLM = 0
```

RANGE OF VALUE: SPRPSDLM takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSDLM contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: Deadlocks among cluster nodes could become a serious problem, so the Lock Manager sends messages to the controlling process whenever a converted lock has been held for more than the SYSGEN parameter allows. If the process is waiting for another lock, it is told to go back to waiting for this resource, and the resource is freed for another job.

S P R P S D M Z | Demand Zero Page Fault Rate

SPRPSDMZ contains the number of demand zero page faults per second during DURATION (measurement interval)

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRDMZFL - Demand Zero Page Faults

ELEMENT DERIVATION: SPRPSDMZ is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSDMZ = SPRDMZFL / DURATION;
  else
    SPRPSDMZ = .;
```

RANGE OF VALUE: SPRPSDMZ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSDMZ contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: Typical values of SPRPSDMZ usually range from 3 to 100, depending on the amount of image activation on the system.

SPRPSDMZ is the number of times per second that a process requested pages of memory to be allocated. For each request, the VMS system initializes the page to zeros. This activity is usually performed during image activation when processes first begin using memory that had been logically allocated at compile time. Therefore, SPRDMZFL is a good estimator of the impact of image activation on paging.

S P R P S D P L | Departing Local Packet Rate

SPRPSDPL contains the number of departing local packets per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRDPLPK - Departing Local Packets

ELEMENT DERIVATION: SPRPSDPL is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSDPL = SPRDPLPK / DURATION
  else
    SPRPSDPL = 0
```

RANGE OF VALUE: SPRPSDPL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSDPL contains the result of the

computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: Departing local packets are DECnet data packets and control messages that originate with this node. The ratio between SPRARTPK and SPRDPLPK gives a rough estimate of the effect of routing on outgoing network communications. If SPRPSDPL is large, then the outgoing communications could become a limit.

```
+-----------------+
| S P R P S D Q I |  Dequeue Incoming Rate
+-----------------+
```

SPRPSDQI contains the number of incoming dequeues per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRDQI - Dequeue Incoming

ELEMENT DERIVATION: SPRPSDQI is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSDQI = SPRDQI / DURATION
  else
    SPRPSDQI = 0
```

RANGE OF VALUE: SPRPSDQI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSDQI contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRPSDQI is the rate per second at which dequeues that originated on the remote nodes were released on the recording node.

```
+-----------------+
| S P R P S D Q L |  Dequeue Local Rate
+-----------------+
```

SPRPSDQL contains the number of local dequeues per second that originated on the recording node and were released on the remote node.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRDQL - Dequeue Local

ELEMENT DERIVATION: SPRPSDQL is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSDQL = SPRDQL / DURATION
  else
    SPRPSDQL = 0
```

RANGE OF VALUE: SPRPSDQL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSDQL contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: None

```
+-----------------+
| S P R P S D Q O |  Dequeue Outgoing Rate
+-----------------+
```

SPRPSDQO contains the number of outgoing dequeues per second that originated on the recording node and released on remote nodes.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRDQO - Dequeue Outgoing

ELEMENT DERIVATION: SPRPSDQO is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSDQO = SPRDQO / DURATION
  else
    SPRPSDQO = 0
```

RANGE OF VALUE: SPRPSDQO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSDQO contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: None

```
+-----------------+
| S P R P S D S R |  Deadlock Search Rate
+-----------------+
SPRPSDSR contains the number of deadlock searches the Lock
Manager performed per second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRDEADS - Deadlock Searches

ELEMENT DERIVATION:  SPRPSDSR is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSDSR = SPRDEADS / DURATION
  else
    SPRPSDSR = 0

RANGE OF VALUE:  SPRPSDSR takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSDSR contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  Deadlocks are a potential problem any time
processes must serialize resources to accomplish their tasks.
Whenever a process holds a resource beyond the time of the
system deadlock wait, the Lock Manager searches all processes
waiting on resources to determine if a deadlock condition
exists.  SPRDEADS is the number of times the Lock Manager
searched for deadlocks.  SPRPSDSR is the number of times per
second the Lock Manager searched for deadlocks.


+-----------------+
| S P R P S E C I |  Enqueue Conversion Incoming Rate
+-----------------+
SPRPSECI contains the number of incoming converted locks per
second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRECI - Enqueue Conversions Incoming

ELEMENT DERIVATION:  SPRPSECI is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSECI = SPRECI / DURATION
  else
    SPRPSECI = 0

RANGE OF VALUE:  SPRPSECI takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSECI contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSECI is the rate per second at which
enqueues that originated by or conversions requested by
remote nodes are granted on the recording node.


+-----------------+
| S P R P S E C L |  Enqueue Conversion Local Rate
+-----------------+
SPRPSECL contains the number of local enqueue conversions per
second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRECL - Enqueue Conversions Local

ELEMENT DERIVATION:  SPRPSECL is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSECL = SPRECL / DURATION
  else
    SPRPSECL = 0

RANGE OF VALUE:  SPRPSECL takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSECL contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSECL is the rate per second at which
enqueues that originated by or conversions requested by the
recording node are granted on the recording node.
```

```
+-----------------+
| S P R P S E C O |  Enqueue Conversion Outgoing Rate
+-----------------+
SPRPSECO contains the number of outgoing enqueue conversions
per second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRECO - Enqueue Conversions Outgoing

ELEMENT DERIVATION:  SPRPSECO is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSECO = SPRECO / DURATION
  else
    SPRPSECO = 0

RANGE OF VALUE:  SPRPSECO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSECO contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSECO is the rate per second at which
enqueues that originated by or conversions requested by the
recording node are granted on remote nodes.


+-----------------+
| S P R P S E I O |  Erase I/O Rate
+-----------------+
SPRPSEIO contains the number of erase I/O operations per
second during DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRERASE - Erase I/O Operations

ELEMENT DERIVATION:  SPRPSEIO is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSEIO = SPRERASE / DURATION
  else
    SPRPSEIO = .

RANGE OF VALUE:  SPRPSEIO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSEIO contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  If SPRPSEIO is high and system I/O performance
is poor, high-water marking should be turned off on the
volumes affected.  By default, VMS disks are initialized with
high-water marking enabled.  The high-water marking feature
is typically used for security purposes because it triggers
the file system to write zeros to all allocated bytes when a
file is deleted.  Although good for security, it could
noticeably affect I/O performance on disks where files are
frequently created and deleted.


+-----------------+
| S P R P S E N Q |  Enqueues Not Queued Rate
+-----------------+
SPRPSENQ contains the number of enqueues not queued per
second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  SPRPSENQ is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSENQ = SPRENQNQ / DURATION
  else
    SPRPSENQ = 0

RANGE OF VALUE:  SPRPSENQ takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSENQ contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  When a process requests an enqueue of a
resource, it can specify what to do if the resource is
already locked; it can wait until the resource becomes
available or it can try again later.  If the request
specified it does not want to wait, the enqueue request will
not be queued.  The occurrence is recorded in SPRENQNQ.  If
SPRENQNQ is consistently large, there may be at least one
process using program logic to determine when the resource
```

becomes available, rather than using the conventional 'wait' option when requesting the enqueue. This can quickly consume CPU time and should be discouraged. The value in SPRPSENQ is the number of times per second this is happening. If SPRPSENQ is greater than zero, the locking behavior of the applications should be investigated.

S P R P S E W T | Enqueue Waits Rate

SPRPSEWT contains the number of enqueue waits per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRENQWT - Enqueue Waits

ELEMENT DERIVATION: SPRPSEWT is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSEWT = SPRENQWT / DURATION
  else
    SPRPSEWT = 0
```

RANGE OF VALUE: SPRPSEWT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSEWT contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When a process requests an enqueue of a resource, it can specify what to do if the resource is already locked; it can wait until the resource becomes available or it can try again later. If the request specified to wait, it will be placed on a queue to wait for the resource to become available. This occurrence is recorded in SPRENQWT. If SPRENQWT is consistently large, there is a great deal of contention for a limited resource(s). When SPRPSEWT is greater than two on a regular basis, the resources need to be investigated to make them more efficient.

S P R P S E X T | Extent Cache Attempt Rate

SPRPSEXT contains the number of times per second the system attempted to allocate file space in extent cache.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPREXTAT - Extent Cache Attempts

ELEMENT DERIVATION: SPRPSEXT is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSEXT = SPREXTAT / DURATION
  else
    SPRPSEXT = 0
```

RANGE OF VALUE: SPRPSEXT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSEXT contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When space is allocated for a file, the first place searched is the extent cache. The search occurrence is recorded in SPREXTAT. If the value in SPRPSEXT is large, the amount of file space allocation may be excessive.

S P R P S F C B | Directory FCB Cache Attempt Rate

SPRPSFCB contains the number of times per second the system attempted to locate a file control block (FCB) in the directory FCB cache.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRFCBAT - Directory FCB Cache Attempts

ELEMENT DERIVATION: SPRPSFCB is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSFCB = SPRFCBAT / DURATION
  else
    SPRPSFCB = 0
```

RANGE OF VALUE: SPRPSFCB takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSFCB contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When a file is opened, created, deleted, extended, or truncated, the system has to find the directory containing the file name. The directory FCB cache is searched for the FCB. The search occurrence is recorded in SPRFCBAT. If the value in SPRPSFCB is large, the application's file activity should be investigated.

S P R P S F F L — FCP Page Fault Rate

SPRPSFFL contains the number of page faults per second during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRFCPFL - FCP Page Faults

ELEMENT DERIVATION: SPRPSFFL is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSFFL = SPRFCPFL / DURATION
  else
    SPRPSFFL = .
```

RANGE OF VALUE: SPRPSFFL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSFFL contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: If SPRPSFFL is large, the file system processing is being held up by paging. Working set sizes should be set so that SPRPSFFL remains relatively small.

S P R P S F H C — File Header Cache Attempt Rate

SPRPSFHC contains the number of times per second the system attempted to locate a file header in the file header cache.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRFHDAT - File Header Cache Attempts

ELEMENT DERIVATION: SPRPSFHC is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSFHC = SPRFHDAT / DURATION
  else
    SPRPSFHC = 0
```

RANGE OF VALUE: SPRPSFHC takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSFHC contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When an open, close, extend, delete, or truncate operation is requested on a file, the file header is located by searching the file header cache. The search occurrence is recorded in SPRFHDAT. If the value in SPRFHDAT is large, file activity should be investigated.

S P R P S F I D — File ID Cache Attempt Rate

SPRPSFID contains the number of times per second the system attempted to obtain file IDs from and return file IDs to the file ID cache.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRFIDAT - File ID Cache Attempts

ELEMENT DERIVATION: SPRPSFID is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSFID = SPRFIDAT / DURATION
  else
```

```
    SPRPSFID = 0

RANGE OF VALUE:  SPRPSFID takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSFID contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  File ID numbers are assigned when files are
created and returned on deletion.  When a file is created,
the file ID cache is searched for the next file ID to use.
This search is recorded in SPRFIDAT.  When a file is deleted,
its file ID is returned to the file ID cache.  The return
occurrence is recorded in SPRFIDAT.  If the value in SPRPSFID
is large, files are being created and deleted.  Heavy amounts
of file creations and deletions lead to disk fragmentation.
This can severely affect performance.


+-----------------+
| S P R P S F L T |  Page Fault Rate
+-----------------+
SPRPSFLT contains the number of page faults per second for
DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRPGFLT - Page Faults
           SPRPSPFL - PGE Page Fault Rate
           SPRPSFFL - FCP Page Fault Rate
           SPRPSIFL - IO Page Fault Rate

ELEMENT DERIVATION:  SPRPSFLT is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSFLT = SPRPGFLT / DURATION
  else
    SPRPSFLT = .

RANGE OF VALUE:  SPRPSFLT takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSFLT contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSFLT is the rate of all the page faults,
except system page faults, on the VMS system.  If SPRPSFLT is
near the number of transactions that can be handled per
second for the disk containing the page file and there is
only one page file, system performance could be severely
affected by the speed of the disk.  When space is available,
it is usually best to have more than one page file and those
files be on separate disks.  Depending on the speed of the
VAX CPU, the rate of page faults may present a CPU bottleneck
as well.

If the page fault rate is consistently high, the parameters
that control working set size should be adjusted so that the
processes doing the most faulting can have working sets large
enough to reduce the paging.  If the high rate of faulting is
due to image activation, perhaps the applications can be
redesigned so that they do not invoke as many images.  High
page faulting degrades system performance.  To reduce page
faulting, the VAX system manager should accordingly adjust
the user authorization file parameters WSDEFAULT, WSQUOTA,
and WSEXTENT along with the SYSGEN parameters PFRATH,
BORROWLIM, and WSINC.

WSDEFAULT, WSQUOTA, and WSEXTENT describe the number of
physical pages allocated to a process.  WSDEFAULT is the
amount with which the process starts.  WSQUOTA is the amount
of memory allowed for the process, regardless of the memory
usage on the system.  WSEXTENT is the limit to the amount of
memory the process may borrow when there is memory to spare.

PFRATH is the number of page faults per 10-second interval
that a process must exceed to be eligible to borrow free
memory.  BORROWLIM is the amount of free memory that must be
available before any will be loaned.  WSINC is the number of
memory pages that are loaned with each increase.


+-----------------+
| S P R P S F N I |  Directory Functions Incoming Rate
+-----------------+
SPRPSFNI contains the number of incoming directory functions.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRFNI - Directory Functions Incoming

ELEMENT DERIVATION:  SPRPSFNI is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSFNI = SPRFNI / DURATION
```

```
  else
    SPRPSFNI = 0
```

RANGE OF VALUE: SPRPSFNI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSFNI contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: Cluster-wide directory functions, such as file creation and deletion, require a lot of coordination on the part of the Lock Manager. SPRPSFNI keeps track of the rate that directory functions from remote nodes are requested.

S P R P S F N O — Directory Functions Outgoing Rate

SPRPSFNO contains the number of outgoing directory functions per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRFNO - Directory Functions Outgoing

ELEMENT DERIVATION: SPRPSFNO is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSFNO = SPRFNO / DURATION
  else
    SPRPSFNO = 0
```

RANGE OF VALUE: SPRPSFNO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSFNO contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: Cluster-wide directory functions, such as file creation and deletion, require a lot of coordination on the part of the Lock Manager. SPRPSFNO keeps track of the rate of those requested from remote nodes.

S P R P S F O P — Files Opened Rate

SPRPSFOP contains the number of files opened per second for DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRFILOP - Files Opened
SPRPSOPN - File Open Rate

ELEMENT DERIVATION: SPRPSFOP is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSFOP = SPRFILOP / DURATION
  else
    SPRPSFOP = .
```

RANGE OF VALUE: SPRPSFOP takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSFOP contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: File opening can consume a lot of resources, both in CPU time and the number of disk I/Os required. If the directory of the file is cached in memory, the resource requirement is less. When a file is opened, the location is looked up in the directory. If the directory is not in cache, the directory must be read from disk into cache. Once the location of the file is found, the file header is then read into the file header cache and the file's table of block locations and index buffers are allocated and read from disk.

If SPRPSFOP is large, a lot of the machine resources are being used to initialize files. To reduce the resources, the VAX system manager can do one of two things: increase directory caches, which takes away user memory, or modify applications so they do not open as many files.

S P R P S F P L — Free List Page Fault Rate

SPRPSFPL contains the number of pages allocated per second from the free page list to satisfy process requests.

TYPE: Computed using specific expression statements -

Numeric, measurement unit is integer.

SEE ALSO: SPRFPGFL - Free List Page Faults

ELEMENT DERIVATION: SPRPSFPL is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSFPL = SPRFPGFL / DURATION;
  else
    SPRPSFPL = .;
```

RANGE OF VALUE: SPRPSFPL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSFPL contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRPSFPL represents the number of pages needed per second from the free page list. Free pages are requested during image activation and when a process needs to expand its working set size. SPRPSFPL is usually proportional to the image activation rate on the system. When the free page list is depleted, the process requesting a free page is put into free page wait state until a free page is available.

Free pages are reclaimed at process termination or when working sets are trimmed.

S P R P S F R I | FCP Disk Read I/O Rate

SPRPSFRI contains the number of read I/Os per second during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRFCPRI - FCP Disk Read I/Os

ELEMENT DERIVATION: SPRPSFRI is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSFRI = SPRFCPRI / DURATION else
    SPRPSFRI = .
```

RANGE OF VALUE: SPRPSFRI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSFRI contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: If SPRPSFRI plus SPRPSFWI is close to the sum of the DSKPSIO for all disks on the node, file system caching is not working right. To improve this, check for fragmentation on the file system and increase the size of the ancillary control process (ACP) cache buffers. To set the ACP cache buffers, modify the SYSGEN parameters that begin with "ACP_".

S P R P S F W I | FCP Disk Write I/O Rate

SPRPSFWI contains the number of write I/Os per second during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRFCPWI - FCP Disk Write I/Os

ELEMENT DERIVATION: SPRPSFWI is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSFWI = SPRFCPWI / DURATION
  else
    SPRPSFWI = .
```

RANGE OF VALUE: SPRPSFWI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSFWI contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: If SPRPSFRI plus SPRPSFWI is close to the sum of the DSKPSIO for all disks on the node, file system caching is not working right. To improve this, check for fragmentation on the file system and increase the size of the ancillary control process (ACP) cache buffers. To set the ACP cache buffers, modify the SYSGEN parameters that begin with "ACP_".

```
+-----------------+
| S P R P S G B L |  Global Valid Page Fault Rate
+-----------------+
SPRPSGBL contains the number of page faults per second that
were served by mapping shared memory that was already valid.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRGBLFL - Global Valid Page Faults

ELEMENT DERIVATION:  SPRPSGBL is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSGBL = PGEGBL / DURATION;
  else
    SPRPSGBL = .;

RANGE OF VALUE:  SPRPSGBL takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSGBL contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSGBL usually ranges from 0 to about 10,
depending on the activation of shared images.

Global valid faults are the least expensive of the page
faulting types.  If SPRPSGBL is consistently above 10, it
indicates a correspondingly high level of activation of
images that have shared code or have a good deal of
interprocess communications.  It also can indicate a large
amount of swapping of active processes, which is the least
efficient form of swapping.  So even though global valid
faults do not represent much of a system impediment, SPRPSGBL
can signal that other problems may be negatively affecting
system performance.


+-----------------+
| S P R P S I F L |  IO Page Fault Rate
+-----------------+
SPRPSIFL contains the number of page faults per second for
DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRIOVFL - IO Page Faults
            SPRPSPFL - PGE Page Fault Rate
            SPRPSFFL - FCP Page Fault Rate
            SPRPSFLT - Page Fault Rate

ELEMENT DERIVATION:  SPRPSIFL is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSIFL = SPRIOVFL / DURATION
  else
    SPRPSIFL = .

RANGE OF VALUE:  SPRPSIFL takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSIFL contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSIFL is the rate of all the page faults,
except system page faults, on the VMS system.  If SPRPSIFL is
near the number of I/O transactions that can be handled per
second for the disk containing the page file, and there is
only one page file, system performance could be severely
affected by the speed of the disk.  When space is available,
it is usually best to have more than one paging file and to
locate those files on separate disks.  Depending on the speed
of the VAX CPU, the rate of page faults may present a CPU
bottleneck because of the CPU time required to perform
paging.

If the page fault rate is consistently high, the parameters
that control working set size should be adjusted so that the
processes doing the most faulting can have working sets large
enough to reduce the paging.  If the high rate of faulting is
due to image activation, perhaps the applications can be
redesigned so that they do not invoke as many images.  High
page faulting degrades system performance.  To reduce page
faulting, the VAX system manager should accordingly adjust
the user authorization file parameters WSDEFAULT, WSQUOTA,
and WSEXTENT along with the SYSGEN parameters PFRATH,
BORROWLIM, and WSINC.

WSDEFAULT, WSQUOTA, and WSEXTENT describe the number of
physical pages allocated to a process.  WSDEFAULT is the
amount with which the process starts.  WSQUOTA is the amount
of memory allowed for the process, regardless of the memory
usage on the system.  WSEXTENT is the limit to the amount of
memory the process may borrow when there is memory to spare.

PFRATH is the number of page faults per 10-second interval
that a process must exceed to be eligible to borrow free
```

memory. BORROWLIM is the amount of free memory that must be available before any will be loaned. WSINC is the number of memory pages that are loaned with each increase.

```
+-----------------+
| S P R P S I N S |  Inswap Rate
+-----------------+
```

SPRPSINS contains the number of processes swapped in per second for DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRINSWP - Inswaps

ELEMENT DERIVATION: SPRPSINS is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSINS = SPRINSWP / DURATION
  else
    SPRPSINS = .
```

RANGE OF VALUE: SPRPSINS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSINS contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When there are a large number of working sets being swapped in, the system performance will degrade. If SPRPSINS is 1 or more consistently, the system may be thrashing--swapping out a process to swap in another, then needing to swap the first process back in immediately. Thrashing usually occurs when the system cannot make any good determination for swapping. Thrashing is a sure signal that the system memory is too small. If SPRPSINS and SPRCOMO are not zero, the system is thrashing on the swap file.

Swapping is a highly I/O intensive because working sets are often very large. Additionally, it can increase CPU usage because of swapper processing, which involves recalculating page offsets, setting up asynchronous system traps for interrupts, and the like. If much swapping, especially to the point of thrashing, is going on, the CPU may show large amounts of idle time. This is because the CPU is blocked from being used while it waits for the much slower I/O subsystem to complete its work.

```
+-----------------+
| S P R P S L C I |  Lock Conversion Incoming Rate
+-----------------+
```

SPRPSLCI contains the number of incoming converted locks per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRLCI - Lock Conversions, Incoming

ELEMENT DERIVATION: SPRPSLCI is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSLCI = SPRLCI / DURATION
  else
    SPRPSLCI = 0
```

RANGE OF VALUE: SPRPSLCI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSLCI contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRLCI counts the rate at which requested locks (enqueues) from remote nodes were actually granted.

```
+-----------------+
| S P R P S L C L |  Lock Conversion Local Rate
+-----------------+
```

SPRPSLCL contains the number of local lock conversions per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRLCL - Lock Conversions, Local

ELEMENT DERIVATION: SPRPSLCL is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSLCL = SPRLCL / DURATION
  else
    SPRPSLCL = 0
```

RANGE OF VALUE: SPRPSLCL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSLCL contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRLCL counts the rate at which requested locks (enqueues) from this node were actually granted.

S P R P S L C O — Lock Conversion Outgoing Rate

SPRPSLCO contains the number of outgoing lock conversions per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRLCO - Lock Conversions, Outgoing

ELEMENT DERIVATION: SPRPSLCO is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSLCO = SPRLCO / DURATION
  else
    SPRPSLCO = 0
```

RANGE OF VALUE: SPRPSLCO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSLCO contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRLCO counts the rate at which requested locks (enqueues) sent to remote nodes were actually granted.

S P R P S L N T — Logical Name Translation Rate

SPRPSLNT contains the number of logical name translations per second for DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRLOGNM - Logical Name Translations

ELEMENT DERIVATION: SPRPSLNT is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSLNT = SPRLOGNM / DURATION
  else
    SPRPSLNT = .
```

RANGE OF VALUE: SPRPSLNT takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSLNT contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: Logical names are translated to parse file names or access other system resources such as disk, tape, or network devices. Translation of logical name happens in kernel mode and is not charged to the process requesting translation. SPRPSLNT is usually proportional to image activation rate because the opening of files and devices for the new image involves logical name translation. If SPRPSLNT is high and MODKERTM is 25 percent or more of MODCPUTM, the applications probably should be restructured to not activate as many images. One key to reducing image activation is to reduce the use of DCL command files on the system.

S P R P S L O S — Transit Packet Loss Rate

SPRPSLOS contains the rate of transit packet loss per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRPKLOS - Transit Packets Lost

ELEMENT DERIVATION: SPRPSLOS is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSLOS = SPRPKLOS / DURATION
  else
    SPRPSLOS = 0
```

RANGE OF VALUE: SPRPSLOS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSLOS contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When the network has a high volume of traffic, packets may be lost during transmission. DECnet provides the capability to recognize when a packet sent from one node is not received by another node. When this occurs, the packet is retransmitted to the original node destination. SPRPKLOS contains the number of packets retransmitted because the network could not process all the incoming packets.

S P R P S M B X | Mailbox Write Rate

SPRPSMBX contains the number of mailbox writes per second for DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRMAILW - Mailbox Writes

ELEMENT DERIVATION: SPRPSMBX is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSMBX = SPRMAILW / DURATION
  else
    SPRPSMBX = .
```

RANGE OF VALUE: SPRPSMBX takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSMBX contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: Mailboxes are mechanisms used for interprocess communication on VAX/VMS. When a process wants to receive messages from other processes, it sets up the mechanism to receive messages and then registers the mailbox with VMS, that places it on a list of mailboxes available. Processes wishing to send messages to that mailbox tell VMS where the message is and the mailbox to which it should be delivered. If the mailbox has room, VMS delivers the message and sends a low-priority interrupt to the receiving process. If the mailbox is full, VMS puts the message on a very high priority interrupt queue that continually checks to see if the mailbox has room. If there are many undeliverable messages on the queue, these high priority interrupts can block other, perhaps more important, interrupts from being processed in a timely manner. These interrupts are processed in kernel mode.

When SPRPSMBX is high, there is a higher likelihood of the mailbox full condition. If SPRPSMBX is high and MODKERTM is 25 percent or more of MODCPUTM then the amount of mailbox writing is potentially harmful.

S P R P S M P L | Modified List Page Fault Rate

SPRPSMPL contains the number of pages allocated per second from the modified page list in DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRMPGFL - Modified List Page Faults

ELEMENT DERIVATION: SPRPSMPL is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSMPL = SPRMPGFL / DURATION
  else
    SPRPSMPL = .
```

RANGE OF VALUE: SPRPSMPL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSMPL contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: Modified pages are those that have been written to the Page file by the memory manager but have not been needed for another process yet. SPRPSMPL is the number of times per second that processes requested pages that were written out before they were needed for the free page list or for another process.

Modified page list faults are among the best kind of page faults to have because no I/O is required, just memory mapping and status changing. The system tries to keep as many modified pages on the list as it can, but if the free list is depleted, it takes modified pages before it begins to trim working sets or to swap processes.

```
+-----------------+
| S P R P S N E I |  New Enqueue Incoming Rate
+-----------------+
SPRPSNEI contains the number of incoming enqueues per second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRNEI - New Enqueue Incoming

ELEMENT DERIVATION:  SPRPSNEI is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSNEI = SPRNEI / DURATION
  else
    SPRPSNEI = 0

RANGE OF VALUE:  SPRPSNEI takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSNEI contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSNEI is the rate per second at which
enqueues that originated on the remote node were performed on
the recording node.


+-----------------+
| S P R P S N E L |  New Enqueue Local Rate
+-----------------+
SPRPSNEL contains the number of local enqueues per second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRNEL - New Enqueue Local

ELEMENT DERIVATION:  SPRPSNEL is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSNEL = SPRNEL / DURATION
  else
    SPRPSNEL = 0

RANGE OF VALUE:  SPRPSNEL takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSNEL contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSNEL is the rate per second at which
enqueues that originated on the recording node were performed
on the recording node.


+-----------------+
| S P R P S N E O |  New Enqueue Outgoing Rate
+-----------------+
SPRPSNEO contains the number of outgoing enqueues per second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRNEO - New Enqueue Outgoing

ELEMENT DERIVATION:  SPRPSNEO is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSNEO = SPRNEO / DURATION
  else
    SPRPSNEO = 0

RANGE OF VALUE:  SPRPSNEO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSNEO contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSNEO is the rate per second at which
enqueues that originated on the recording node are performed
on remote nodes.


+-----------------+
| S P R P S N E Q |  New Enqueue Rate
+-----------------+
SPRPSNEQ contains the number of new enqueues per second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.
```

SEE ALSO: Not applicable

ELEMENT DERIVATION: SPRPSNEQ is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSNEQ = SPRNENQ/DURATION
  else
    SPRPSNEQ = 0
```

RANGE OF VALUE: SPRPSNEQ takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSNEQ contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When a process requests a resource via an enqueue, SPRNENQ is incremented. If the resource is unavailable, one of SPRENQNQ or SPRENQWT will be incremented. Enqueues that result in immediate locks can be determined by subtracting SPRENQNQ and SPRENQWT from SPRNENQ. SPRPSNEQ is the rate that resources are being requested.

S P R P S N F L — New Files Rate

SPRPSNFL contains the number of new files per second during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRNEWFL - New Files
SPRPSALL - Disk Allocation Rate
SPRPSACC - File Access Rate

ELEMENT DERIVATION: SPRPSNFL is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSNFL = SPRNEWFL / DURATION
  else
    SPRPSNFL = .
```

RANGE OF VALUE: SPRPSNFL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSNFL contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When a file is created, the file system determines if there is a requested allocation size; if not, the default size is used. The directory is scanned to determine whether a file of the same name is already present. If the file exists, the new file is given a version number one higher than the existing file. The name of the file is added to the directory, the file space is allocated, and the new directory entry is updated to point at the allocated file. If the directory is in cache, these activities require only two physical I/Os: allocating the space and writing out the directory on completion.

If SPRPSNFL is high, check to see if the system is spending a lot of CPU time on the file system. In addition, a high value in SPRPSNFL should be a signal to check for disk fragmentation. If SPRPSNFL and SPRPSTRN are both high, the system may be thrashing on the file map.

S P R P S N L I — New Locks Incoming Rate

SPRPSNLI contains the number of incoming enqueues per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRNLI - New Locks, Incoming

ELEMENT DERIVATION: SPRPSNLI is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSNLI = SPRNLI / DURATION
  else
    SPRPSNLI = 0
```

RANGE OF VALUE: SPRPSNLI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSNLI contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRNLI counts the number of requested locks (enqueues) received per second from remote nodes.

```
+-----------------+
| S P R P S N L L |  New Locks Local Rate
+-----------------+
SPRPSNLL contains the number of local enqueues per second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRNLL - New Locks, Local

ELEMENT DERIVATION:  SPRPSNLL is computed using the following
algorithm:
  if DURATION > 0 then
    SPRPSNLL = SPRNLL / DURATION
  else
    SPRPSNLL = 0

RANGE OF VALUE:  SPRPSNLL takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSNLL contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSNLL counts the number of requested locks
(enqueues) per second from this node that were newly queued.


+-----------------+
| S P R P S N L O |  New Locks Outgoing Rate
+-----------------+
SPRPSNLO contains the number of outgoing enqueues per second.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRNLO - New Locks, Outgoing

ELEMENT DERIVATION:  SPRPSNLO is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSNLO = SPRNLO / DURATION
  else
    SPRPSNLO = 0

RANGE OF VALUE:  SPRPSNLO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSNLO contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSNLO counts the number of requested locks
(enqueues) per second that were newly sent to remote nodes.



+-----------------+
| S P R P S O P N |  File Open Rate
+-----------------+
SPRPSOPN contains the number of files opened per second
during DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPROPEN - Files Opened

ELEMENT DERIVATION:  SPRPSOPN is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSOPN = SPROPEN / DURATION
  else
    SPRPSOPN = .

RANGE OF VALUE:  SPRPSOPN takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSOPN contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  Opening files can consume a lot of resources,
both in CPU time and the number of disk I/Os required.  If
the directory of the file is cached in memory, the resource
requirement is less.  When a file is opened, the location is
looked up in the directory.  If the directory is not in
cache, the directory must first be read from disk into cache.
Once the location of the file is found, the file header is
then read into the file header cache, and the file's table of
block locations and index buffers are allocated and read from
disk.

If SPRPSOPN is large, a lot of the machine resources are
being used to initialize the use of files.  If this seems to
be affecting performance, the VAX system manager can do one
of two things:  increase directory caches, which takes away
user memory, or modify applications so they do not open as
many files.
```

```
+-----------------+
| S P R P S P F L |  PGE Page Fault Rate
+-----------------+
SPRPSPFL contains the number of pages allocated per second to
processes as a result of page faults during DURATION
(measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRPGEFL - PGE Page Faults
           SPRPSIFL - IO Page Fault Rate
           SPRPSFFL - FCP Page Fault Rate
           SPRPSFLT - Page Fault Rate

ELEMENT DERIVATION:  SPRPSPFL is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSPFL = SPRPGEFL / DURATION;
  else
    SPRPSPFL = .;

RANGE OF VALUE:  SPRPSPFL takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSPFL contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSPFL is the rate of all the page faults,
except system page faults, on the VMS system.  If PGPSFLT is
near the number of transactions that can be handled per
second for the disk with the page file on it and there is
only one page file, system performance could be severely
affected by the speed of the disk.  When space is available,
it is usually best to have more than one paging file on
separate disks.  Depending on the speed of the CPU, the rate
of page faults may present a CPU bottleneck as well.

If the page fault rate is consistently high, the parameters
that control working set size should be adjusted so that the
processes that are doing the most faulting can have working
sets that are large enough to reduce the paging.  If the high
rate of faulting is due to image activation, perhaps the
applications can be redesigned so that they do not invoke as
many images.
```

```
+-----------------+
| S P R P S P G R |  Page Read Rate
+-----------------+
SPRPSPGR contains the number of page reads per second for
DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRPGRD  - Page Reads
           SPRPSRED - PGE Page Read Rate

ELEMENT DERIVATION:  SPRPSPGR is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSPGR = SPRPGRD / DURATION
  else
    SPRPSPGR = .

RANGE OF VALUE:  SPRPSPGR takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSPGR contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSPGR is a count of pages per second read
from disk, not the number of I/O transactions.  VMS tries to
package several page fault reads into one I/O transaction if
possible.  Nevertheless, a high SPRPSPGR is much more harmful
to performance than the fault rates that are satisfied in
memory, such as SPRPSFPL, SPRPSMPL, or SPRPSGBL.
```

```
+-----------------+
| S P R P S P G W |  Page Write Rate
+-----------------+
SPRPSPGW contains the number of page writes per second for
DURATION (measurement interval).

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRPGWT  - Page Writes
           SPRPSWRT - PGE Page Write Rate

ELEMENT DERIVATION:  SPRPSPGW is computed using the following
algorithm:

  if DURATION > 0 then
```

```
    SPRPSPGW = SPRPGWT / DURATION
  else
    SPRPSPGW = .
```

RANGE OF VALUE: SPRPSPGW takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSPGW contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRPSPGW is an indication of heavy page faulting since memory is being written out to disk to make room for other memory to be allocated. If the data in the pages is needed again, those pages will have to be read in once more. Since the memory manager tries to save up the pages to be written, SPRPSPGW is usually larger than SPRPGWT.

S P R P S P R | Page Read I/O Rate

SPRPSPR contains the number of page read I/Os per second for DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRPRDIO - Page Read I/Os
SPRPSPRI - PGE Page Read I/O Rate
SPRPSFRI - FCP Disk Read I/O Rate

ELEMENT DERIVATION: SPRPSPR is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSPR = SPRPRDIO / DURATION
  else
    SPRPSPR = .
```

RANGE OF VALUE: SPRPSPR takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSPR contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When a page fault causes a page to be read from disk, the memory manager looks to see if any other reads can be combined with it to reduce the number of actual I/O transactions. Thus, SPRPSPR can never exceed SPRPSPGR. If SPRPSPR is near 50 percent of the number of I/O transactions that the disks on the system can support, the system is probably experiencing severe disk I/O limitations as well. If this high amount of read I/Os exists, the VMS system parameters must be adjusted to reduce paging or the workload of the system must be reduced.

S P R P S P R I | PGE Page Read I/O Rate

SPRPSPRI contains the number of read I/O operations per second to satisfy page faults for DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRPGERI - PGE Page Read I/Os
SPRPSPR - Page Read I/O Rate
SPRPSFRI - FCP Disk Read I/O Rate

ELEMENT DERIVATION: SPRPSPRI is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSPRI = SPRPGERI / DURATION;
  else
    SPRPSPRI = .;
```

RANGE OF VALUE: SPRPSPRI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSPRI contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When a page fault causes a page to be read from disk, the memory manager determines if any other reads can be combined with it to minimize the number of actual I/O transactions. Thus, SPRPSPRI can never exceed SPRPSRED. If SPRPSPRI is more than 50 percent of the number of I/O transactions that the disk device can support, the system is probably experiencing severe disk I/O limitations as well. If a high amount of read I/Os exists, the VMS system parameters must be adjusted to reduce paging or the workload of the system must be reduced.

```
+-----------------+
| S P R P S P W   |  Page Write I/O Rate
+-----------------+
SPRPSPW contains the number of page write I/Os per second for
DURATION (measurement interval).

TYPE: Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRPWIO  - Page Write I/Os
           SPRPSFWI - FCP Disk Write I/O Rate
           SPRPSPWI - PGE Page Write I/O Rate

ELEMENT DERIVATION: SPRPSPW is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSPW = SPRPWIO / DURATION
  else
    SPRPSPW = .

RANGE OF VALUE: SPRPSPW takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS: SPRPSPW contains the result of the
computation described above in ELEMENT DERIVATION. The value
is recomputed at each level of summarization.

USAGE NOTES: When the memory manager needs to write a page to
the page file, it checks to see if there are other pages it
can write in the same transaction.  Thus, SPRPSPW can never
be larger than SPRPSPGW.  If SPRPSPW plus SPRPSPR is close to
the number of I/O operations supported by the system, system
performance will be severely affected by paging activity.
When there are a large number of page I/Os, steps should be
taken to reduce the paging immediately.  Adjusting working
sets higher or lowering the number of concurrent processes or
both may be necessary to reduce paging to where disk I/O does
not limit system performance.


+-----------------+
| S P R P S P W I |  PGE Page Write I/O Rate
+-----------------+
SPRPSPWI contains the number of write operations per second
to the Page file.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRPGEWI - PGE Page Write I/Os
           SPRPSPW  - Page Write I/O Rate
           SPRPSFWI - FCP Disk Write I/O Rate

ELEMENT DERIVATION:  SPRPSPWI is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSPWI = SPRPGEWI / DURATION;
  else
    SPRPSPWI = .;

RANGE OF VALUE:  SPRPSPWI takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSPWI contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  When the memory manager needs to write a page
to the Page file, it determines if there are other pages it
can write in the same transaction.  Thus, SPRPSPWI can never
be larger than SPRPSWRT.  If SPRPSPWI plus SPRPSPRI is close
to the number of I/O operations supported by the system, the
performance will be severely affected by paging activity.
When there are a large number of page I/Os, steps should be
taken to reduce the paging immediately.  Adjusting working
sets higher or lower or lowering the number of concurrent
processes or both may be necessary to reduce paging to where
disk I/O does not degrade system performance.



+-----------------+
| S P R P S Q O T |  Quota Cache Attempt Rate
+-----------------+
SPRPSQOT contains the number of times per second the system
made a quota cache attempt.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRQUOAT - Quota Cache Attempts

ELEMENT DERIVATION:  SPRPSQOT is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSQOT = SPRQUOAT / DURATION
  else
    SPRPSQOT = 0

RANGE OF VALUE:  SPRPSQOT takes any integer value from 0 to
```

2,147,483,647.

SUMMARIZATION PROCESS: SPRPSQOT contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: If quota checking is enabled, a change in file size causes a search of the quota cache for an entry describing the file owner. This search is recorded in SPRQUOAT. If the value in SPRPSQOT is large, consider disabling quota checking.

S P R P S R B F | Receiver Buffer Failure Rate

SPRPSRBF contains the number of receiver buffer failures per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRBUFFL - Receiver Buffer Failures

ELEMENT DERIVATION: SPRPSRBF is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSRBF = SPRBUFFL / DURATION
  else
    SPRPSRBF = 0
```

RANGE OF VALUE: SPRPSRBF takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSRBF contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When the target node is unable to receive a packet, this node knows it must retransmit the message. SPRBUFFL represents packets that were resent because of buffer problems in the target nodes. If this occurs, the target node is having problems with its nonpaged pool.

S P R P S R E D | PGE Page Read Rate

SPRPSRED contains the number of pages per second read from disk to satisfy page faults for DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRPGERD - PGE Page Reads
SPRPSPGR - Page Read Rate

ELEMENT DERIVATION: SPRPSRED is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSRED = SPRPGERD / DURATION;
  else
    SPRPSRED = .;
```

RANGE OF VALUE: SPRPSRED takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSRED contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRPSRED is a count of pages per second read from disk, not the number of I/O transactions. VMS tries to group multiple page fault reads into one I/O transaction if possible. Nevertheless, a high SPRPSRED is much more harmful to performance than the fault rates that are satisfied in memory, such as SPRPSFPL, SPRPSMPL or SPRPSGBL.

S P R P S R I O | Read I/O Rate

SPRPSRIO contains the number of page read I/Os per second for DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRRDIOS - Page Read I/Os
SPRPSPR - Page Read I/O Rate
SPRPSPRI - PGE Page Read I/O Rate

ELEMENT DERIVATION: SPRPSRIO is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSRIO = SPRRDIOS / DURATION
  else
    SPRPSRIO = .
```

RANGE OF VALUE: SPRPSRIO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSRIO contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When a page fault causes a page to be read from disk, the memory manager looks to see if any other reads can be combined with it to reduce the number of actual I/O transactions. If SPRPSRIO is near 50 percent of the number of I/O transactions that the disks on the system can support, the system is probably experiencing severe disk I/O limitations as well. If this high amount of read I/Os exists, the VMS system parameters must be adjusted to reduce paging or the system workload must be reduced.

S P R P S S P L — Split Transfer Rate

SPRPSSPL contains the number of cache buffer split transfers per second for DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRSPLIT - Split Transfers

ELEMENT DERIVATION: SPRPSSPL is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSSPL = SPRSPLIT / DURATION
  else
    SPRPSSPL = .
```

RANGE OF VALUE: SPRPSSPL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSSPL contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When an I/O request is given to VMS and the cache buffers to receive it are too small, the request will be broken into as many requests as necessary to satisfy the I/O request. The requesting process will not be signaled that the I/O is complete until the last of the split transfers is done. Split transfers will lengthen the I/O response time to the process. If this value is high, consider increasing the size of the cache buffer.

S P R P S S Y S — System Page Fault Rate

SPRPSSYS contains the number of system page faults per second in DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRSYSFL - System Page Faults

ELEMENT DERIVATION: SPRPSSYS is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSSYS = SPRSYSFL / DURATION;
  else
    SPRPSSYS = .;
```

RANGE OF VALUE: SPRPSSYS takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSSYS contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: System page faults are a potential bottleneck to all processes on the system. SPRPSSYS represents the number of times per second that virtual addresses resolve to system space, but the required page is not available in the system working set. Since the system working set services all processes on the system, the likelihood of a page being needed by another process immediately is low. This means that when one process causes a page to be read in, a page for another process was written out. The other process will likely need that page again when it processes, so it will also fault. This chain reaction causes system space thrashing and will quickly degrade system performance. If SPRPSSYS is more than 1 consistently or if it ever peaks above 5, steps should be taken to correct the problem.

System working set size should be increased to the point

where SPRPSSYS is consistently 1 or less. The SYSGEN parameter that controls this is SYSMWCNT. SYSMWCNT should not be too large, however, because that space is taken away from the memory available for process working sets, which also could cause paging and swapping bottlenecks.

S P R P S T A B — Transaction Abort Rate

SPRPSTAB contains the average count per second of transaction operations aborted by DECdtm services. It is the total of planned and unplanned aborts.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRCNTAB - Transactions Aborted Count
SPRMSTAB - Max Transaction Abort Rate

ELEMENT DERIVATION: SPRPSTAB is computed using the following algorithm:

SPRPSTAB = SPRCNTAB / DURATION ;

RANGE OF VALUE: SPRPSTAB ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRPSTAB contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: The Abort Transaction system service can be called by any resource manager or transaction manager participating in the transaction. This might be due to a machine failure, for example. In the abort phase, all transaction participants are ordered to abort and roll back their work.

S P R P S T A D — Transaction Remote Add Rate

SPRPSTAD contains the average count per second of add remote (to a remote subordinate parent) branch operations by DECdtm services.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRCNTAD - Add Remote Branch Ops Count
SPRMSTAD - Max Transaction Remote Add Rate

ELEMENT DERIVATION: SPRPSTAD is computed using the following algorithm:

SPRPSTAD = SPRCNTAD / DURATION ;

RANGE OF VALUE: SPRPSTAD ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRPSTAD contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: None

S P R P S T B R — Transaction Remote Start Rate

SPRPSTBR contains the average count per second of start remote (to a remote parent) branch operations by DECdtm services.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRCNTBR - Start Remote Branch Ops Count
SPRMSTBR - Max Transaction Remote Start Rate

ELEMENT DERIVATION: SPRPSTBR is computed using the following algorithm:

SPRPSTBR = SPRCNTBR / DURATION ;

RANGE OF VALUE: SPRPSTBR ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRPSTBR contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: None

S P R P S T C O — Transaction Total Commit Rate

SPRPSTCO contains the average count per second of transactions committed in total. It is based on the combined count of one-phase and two-phase commit operations completed by DECdtm services.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRCNTCO - Transactions Total Committed Count
SPRMSTCO - Max Transaction Total Commit Rate

ELEMENT DERIVATION: SPRPSTCO is computed using the following algorithm:

```
SPRPSTCO = SPRCNTCO / DURATION ;
```

RANGE OF VALUE: SPRPSTCO ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRPSTCO contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: None

S P R P S T E N | Transaction End Rate

SPRPSTEN contains the average count per second of transaction operations ended by DECdtm services.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRCNTEN - Transactions Ended Count
SPRMSTEN - Max Transaction End Rate

ELEMENT DERIVATION: SPRPSTEN is computed using the following algorithm:

```
SPRPSTEN = SPRCNTEN / DURATION ;
```

RANGE OF VALUE: SPRPSTEN ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRPSTEN contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRPSTEN reflects the rate at which the $END_TRANS system service completed.

S P R P S T O P | Transaction One-Phase Init Rate

SPRPSTOP contains the average count per second of one-phase commit operations initiated by DECdtm services.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRCNTOP - One-Phase Commits Initiated
SPRMSTOP - Max Transaction One-Phase Init Rate

ELEMENT DERIVATION: SPRPSTOP is computed using the following algorithm:

```
SPRPSTOP = SPRCNTOP / DURATION ;
```

RANGE OF VALUE: SPRPSTOP ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRPSTOP contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRPSTOP reflects the rate of transactions processed using a lower resource operation because there is only one resource manager participating in the transaction.

S P R P S T P R | Transaction Prepare Rate

SPRPSTPR contains the average count per second of transactions placed in the prepare state by DECdtm services.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRCNTPR - Transactions Prepared Count
SPRMSTPR - Max Transaction Prepare Rate

ELEMENT DERIVATION: SPRPSTPR is computed using the following algorithm:

```
SPRPSTPR = SPRCNTPR / DURATION ;
```

RANGE OF VALUE: SPRPSTPR ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRPSTPR contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: The prepare phase is the first phase of the two-phase commit protocol implemented by DECdtm services when the $END_TRANS service is called.

S P R P S T R N — Window Turn Rate

SPRPSTRN contains the number of window turns per second during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRTURN - Window Turns

ELEMENT DERIVATION: SPRPSTRN is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSTRN = SPRTURN / DURATION
  else
    SPRPSTRN = .
```

RANGE OF VALUE: SPRPSTRN takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSTRN contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: Window turns occur when the file map is not completely in the system buffers. To access all portions of the file system, multiple pages of the file map need to be read in to find the physical locations of the portions needed. On systems where there are many small files created and deleted, the file system gradually becomes fragmented into small pieces over time. When large files are allocated, they are broken up into several noncontiguous pieces. When the system performs I/O to large fragmented files, system performance can be negatively affected by the amount of I/O time and CPU processing necessary to perform these window turns. Usually when SPRPSTRN is large, the system is thrashing on the file map and performance severely degrades.

S P R P S T S T — Transaction Start Rate

SPRPSTST contains the average count per second of transaction operations started by DECdtm services.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRCNTST - Transactions Started Count
SPRMSTST - Max Transaction Start Rate

ELEMENT DERIVATION: SPRPSTST is computed using the following algorithm:

```
SPRPSTST = SPRCNTST / DURATION ;
```

RANGE OF VALUE: SPRPSTST ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRPSTST contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRPSTST reflects the rate of successful completions of the $START_TRANS system service.

S P R P S U N I — Unlock Incoming Rate

SPRPSUNI contains the number of incoming dequeues per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRUNI - Unlocks, Incoming

ELEMENT DERIVATION: SPRPSUNI is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSUNI = SPRUNI / DURATION
  else
    SPRPSUNI = 0
```

RANGE OF VALUE: SPRPSUNI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSUNI contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRPSUNI counts the rate of release of locks from remote nodes.

S P R P S U N L | Unlock Local Rate

SPRPSUNL contains the number of local dequeues per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRUNL - Unlocks, Local

ELEMENT DERIVATION: SPRPSUNL is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSUNL = SPRUNL / DURATION
  else
    SPRPSUNL = 0
```

RANGE OF VALUE: SPRPSUNL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSUNL contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRUNL counts the rate of release of locks from this node.

S P R P S U N O | Unlock Outgoing Rate

SPRPSUNO contains the number of outgoing dequeues per second.

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRUNO - Unlocks, Outgoing

ELEMENT DERIVATION: SPRPSUNO is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSUNO = SPRUNO / DURATION
  else
    SPRPSUNO = 0
```

RANGE OF VALUE: SPRPSUNO takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSUNO contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: SPRPSUNO counts the rate of release of locks from this node on remote nodes.

S P R P S V L K | Volume Lock Wait Rate

SPRPSVLK contains the number of volume lock waits per second during DURATION (measurement interval).

TYPE: Computed using specific expression statements - Numeric, measurement unit is integer.

SEE ALSO: SPRVLOK - Volume Lock Waits

ELEMENT DERIVATION: SPRPSVLK is computed using the following algorithm:

```
  if DURATION > 0 then
    SPRPSVLK = SPRVLOK / DURATION
  else
    SPRPSVLK = .
```

RANGE OF VALUE: SPRPSVLK takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRPSVLK contains the result of the computation described above in ELEMENT DERIVATION. The value is recomputed at each level of summarization.

USAGE NOTES: When file system operations involve changing the disk volume-wide information, volume access must be serialized to prevent collisions by competing processes. SPRPSVLK is the number of times per second that processes had to wait to access a volume. If SPRPSVLK is large, I/O response could be elongated because of contention for volume access.

```
+-----------------+
| S P R P S W I P |  Write In Progress Page Fault Rate
+-----------------+
SPRPSWIP contains the number of page faults per second that
are satisfied by pages in the process of being written out.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRWIPFL - Write In Progress Page Faults

ELEMENT DERIVATION:  SPRPSWIP is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSWIP = SPRWIPFL / DURATION;
  else
    SPRPSWIP = .;

RANGE OF VALUE:  SPRPSWIP takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSWIP contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSWIP is typically 0 or very close to 0.
This is the rate for an unusual class of page faults, the
pages were marked as removable, and as the memory manager was
writing them out the processes accessed them again.  It will
be rare to see SPRPSWIP greater than 1.  If it is
consistently greater than 0, there is likely a severe problem
with the I/O subsystem.  The VMS error log file should be
checked over several DURATIONs (measurement interval).


+-----------------+
| S P R P S W R T |  PGE Page Write Rate
+-----------------+
SPRPSWRT contains the number of pages per second written to
the Page file.

TYPE:  Computed using specific expression statements -
Numeric, measurement unit is integer.

SEE ALSO:  SPRPGEWR - PGE Page Writes
           SPRPSPGW - Page Write Rate

ELEMENT DERIVATION:  SPRPSWRT is computed using the following
algorithm:

  if DURATION > 0 then
    SPRPSWRT = SPRPGEWR / DURATION;
  else
    SPRPSWRT = .;

RANGE OF VALUE:  SPRPSWRT takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPSWRT contains the result of the
computation described above in ELEMENT DERIVATION.  The value
is recomputed at each level of summarization.

USAGE NOTES:  SPRPSWRT is a sign of hard page faulting since
memory is being written out to disk to make room for other
memory to be allocated.  If the data in the pages is needed
again then those pages will have to be read in once more.
Since the memory manager tries to save up the pages to be
written, SPRPSWRT is usually larger than SPRPSPWI.


+-----------------+
| S P R P W I O   |  Page Write I/Os
+-----------------+
SPRPWIO contains the count of write operations to the page
file to make room for requested pages.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSPW - Page Write I/O Rate
           SPRPGWT  - Page Writes
           SPRPGEWI - PGE Page Write I/Os
           SPRFCPWI - FCP Disk Write I/Os

ELEMENT DERIVATION:  SPRPWIO is set to the value of the
MNR_IO$L_PWRITIO field of the VMS MONITOR IO record.

RANGE OF VALUE:  SPRPWIO takes an integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRPWIO contains the sum of all
values of SPRPWIO processed in the level of summarization.

USAGE NOTES:  When the memory manager needs to write a page
to the page file, it checks to see if there are other pages
it can write in the same transaction.  Thus, SPRPWIO can
never be larger than SPRPGWT.
```

```
+-----------------+
| S P R Q U O A T |  Quota Cache Attempts
+-----------------+
SPRQUOAT contains the number of attempts to find information
in the quota cache.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRQUOHT - Quota Cache Hits

ELEMENT DERIVATION:  SPRQUOAT is set to the value of the
MNR_FIL$L_QUO_TRIES field in the VMS MONITOR
FILE_SYSTEM_CACHE record.

RANGE OF VALUE:  SPRQUOAT takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRQUOAT contains the sum of all
values of SPRQUOAT processed in the level of summarization.

USAGE NOTES:  If quota checking is enabled, a change in file
size causes a search of the quota cache for an entry
describing the file owner.  The search occurrence is recorded
in SPRQUOAT.  If the entry is found, the occurrence is
recorded in SPRQUOHT.  If SPRQUOAT is much larger than
SPRQUOHT, quota cache is not large enough and quota checking
could be causing performance problems.  The size of the quota
cache is set by the SYSGEN ACP_QUOCACHE.


+-----------------+
| S P R Q U O H T |  Quota Cache Hits
+-----------------+
SPRQUOHT contains the number of times the system successfully
found an entry in the quota cache.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRQUOAT - Quota Cache Attempts

ELEMENT DERIVATION:  SPRQUOHT is set to the value of the
MNR_FIL$L_QUO_HIT field of the VMS MONITOR FILE_SYSTEM_CACHE
record.

RANGE OF VALUE:  SPRQUOHT takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRQUOHT contains the sum of all
values of SPRQUOHT processed in the level of summarization.

USAGE NOTES:  If quota checking is enabled, a change in file
size causes a search of the quota cache for an entry
describing the file owner.  This search occurrence is
recorded in SPRQUOAT.  If the entry is found, the occurrence
is recorded in SPRQUOHT.  If SPRQUOAT is much larger than
SPRQUOHT, quota cache is not large enough and quota checking
could be causing performance problems.  The size of the quota
cache is set by the SYSGEN ACP_QUOCACHE.


+-----------------+
| S P R R D I O S |  Page Read I/Os
+-----------------+
SPRRDIOS contains the count of read I/O operations from disk
to satisfy page faults for DURATION (measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRPSRIO - Read I/O Rate
           SPRPRDIO - Page Read I/Os
           SPRPGERI - PGE Page Read I/Os

ELEMENT DERIVATION:  SPRRDIOS is set to the value of the
MNR_SYS$L_PREADIO field of the VMS MONITOR SYSTEM record.

RANGE OF VALUE:  SPRRDIOS takes an integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRRDIOS contains the sum of all
values of SPRRDIOS processed in the level of summarization.

USAGE NOTES:  When a page fault causes a page to be read from
disk, the memory manager looks to see if any other reads can
be combined with it to reduce the number of actual I/O
transactions.  To improve efficiency it may be necessary to
adjust some file caching parameters.


+-----------------+
| S P R R E Q S T |  MSCP Requests
+-----------------+
SPRREQST contains the count of requests for I/O by remote
processors.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRMSCRD - MSCP Reads
```

SPRMSCWR - MSCP Writes

ELEMENT DERIVATION: SPRREQST is set to the value of the MNR_MSC$L_REQUEST field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRREQST takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRREQST contains the sum of all values of SPRREQST processed in the level of summarization.

USAGE NOTES: SPRREQST is the total number of I/O requests by remote processors. As such, SPRREQST should equal SPRMSCRD plus SPRMSCWR. It should also equal the sum of SPRSIZE1 through SPRSIZE7. To see the impact of MSCP server I/Os on the I/O subsystem, compare SPRREQST plus SPRFRGMT to SPRDIOS.

S P R S I Z E 1 — 1 Block I/Os

SPRSIZE1 contains the number of 1 block I/Os requested by the remote processors.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRREQST - MSCP Requests
SPRMSCSP - MSCP Splits

ELEMENT DERIVATION: SPRSIZE1 is set to the value of the MNR_MSC$L_SIZE1 field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRSIZE1 takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSIZE1 contains the sum of all values of SPRSIZE1 processed in the level of summarization.

USAGE NOTES: MSCP request sizes cause several trade-offs. Small requests are less likely to be fragmented, but the ratio of overhead bytes in the transfer protocol is worse, as is the ratio of disk latency to data transferred. Large requests are more efficient, but if buffer space is scarce, the likelihood of fragmentation is greater.

S P R S I Z E 2 — 2-3 Block I/Os

SPRSIZE2 contains the number of 2-3 block I/Os requested by the remote processors.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRREQST - MSCP Requests
SPRMSCSP - MSCP Splits

ELEMENT DERIVATION: SPRSIZE2 is set to the value of the MNR_MSC$L_SIZE2 field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRSIZE2 takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSIZE2 contains the sum of all values of SPRSIZE2 processed in the level of summarization.

USAGE NOTES: MSCP request sizes cause several trade-offs. Small requests are less likely to be fragmented, but the ratio of overhead bytes in the transfer protocol is worse, as is the ratio of disk latency to data transferred. Large requests are more efficient, but if buffer space is scarce, the likelihood of fragmentation is greater.

S P R S I Z E 3 — 4-7 Block I/Os

SPRSIZE3 contains the number of 4-7 block I/Os requested by the remote processors.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRREQST - MSCP Requests
SPRMSCSP - MSCP Splits

ELEMENT DERIVATION: SPRSIZE3 is set to the value of the MNR_MSC$L_SIZE3 field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRSIZE3 takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSIZE3 contains the sum of all values of SPRSIZE3 processed in the level of summarization.

USAGE NOTES: MSCP request sizes cause several trade-offs.

Small requests are less likely to be fragmented, but the ratio of overhead bytes in the transfer protocol is worse, as is the ratio of disk latency to data transferred. Large requests are more efficient, but if buffer space is scarce, the likelihood of fragmentation is greater.

S P R S I Z E 4 | 8-15 Block I/Os

SPRSIZE4 contains the number of 8-15 block I/Os requested by the remote processors.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRREQST - MSCP Requests
SPRMSCSP - MSCP Splits

ELEMENT DERIVATION: SPRSIZE4 is set to the value of the MNR_MSC$L_SIZE4 field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRSIZE4 takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSIZE4 contains the sum of all values of SPRSIZE4 processed in the level of summarization.

USAGE NOTES: MSCP request sizes cause several trade-offs. Small requests are less likely to be fragmented, but the ratio of overhead bytes in the transfer protocol is worse, as is the ratio of disk latency to data transferred. Large requests are more efficient, but if buffer space is scarce, the likelihood of fragmentation is greater.

S P R S I Z E 5 | 16-31 Block I/Os

SPRSIZE5 contains the number of 16-31 block I/Os requested by the remote processors.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRREQST - MSCP Requests
SPRMSCSP - MSCP Splits

ELEMENT DERIVATION: SPRSIZE5 is set to the value of the MNR_MSC$L_SIZE5 field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRSIZE5 takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSIZE5 contains the sum of all values of SPRSIZE5 processed in the level of summarization.

USAGE NOTES: MSCP request sizes cause several trade-offs. Small requests are less likely to be fragmented, but the ratio of overhead bytes in the transfer protocol is worse, as is the ratio of disk latency to data transferred. Large requests are more efficient, but if buffer space is scarce, the likelihood of fragmentation is greater.

S P R S I Z E 6 | 32-63 Block I/Os

SPRSIZE6 contains the number of 32-63 block I/Os requested by the remote processors.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRREQST - MSCP Requests
SPRMSCSP - MSCP Splits

ELEMENT DERIVATION: SPRSIZE6 is set to the value of the MNR_MSC$L_SIZE6 field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRSIZE6 takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSIZE6 contains the sum of all values of SPRSIZE6 processed in the level of summarization.

USAGE NOTES: MSCP request sizes cause several trade-offs. Small requests are less likely to be fragmented, but the ratio of overhead bytes in the transfer protocol is worse, as is the ratio of disk latency to data transferred. Large requests are more efficient, but if buffer space is scarce, the likelihood of fragmentation is greater.

S P R S I Z E 7 | 64 And Larger Block I/Os

SPRSIZE7 contains the number of 64 block and larger I/Os requested by the remote processors.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRREQST - MSCP Requests
SPRMSCSP - MSCP Splits

ELEMENT DERIVATION: SPRSIZE7 is set to the value of the MNR_MSC$L_SIZE7 field of the VMS MONITOR MSCP_SERVER record.

RANGE OF VALUE: SPRSIZE7 takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSIZE7 contains the sum of all values of SPRSIZE7 processed in the level of summarization.

USAGE NOTES: MSCP request sizes cause several trade-offs. Small requests are less likely to be fragmented, but the ratio of overhead bytes in the transfer protocol is worse, as is the ratio of disk latency to data transferred. Large requests are more efficient, but if buffer space is scarce, the likelihood of fragmentation is greater.

S P R S M B L K — Block Less Than or Equal to 32

SPRSMBLK contains the number of blocks less than or equal to 32 bytes in size.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMXSMB - Maximum Block Less Than or Equal to 32

ELEMENT DERIVATION: SPRSMBLK is set to the value of the MNR_POO$L_SMALLCNT field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRSMBLK takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSMBLK contains the last value of SPRSMBLK processed in the level of summarization.

USAGE NOTES: As the dynamic pool gets fragmented, little blocks are formed. Typically, most requests are for more than 32 bytes, so the number held in SPRSMBLK is the number of relatively useless blocks on the dynamic pool.

S P R S M H L — Smallest Block Unused

SPRSMHL contains the size (in bytes) of the smallest block in the pool.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMNSBK- Minimum Smallest Block

ELEMENT DERIVATION: SPRSMHL is set to the value of the MNR_POO$L_SMALLHOLE field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRSMHL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSMHL contains the last value of SPRSMHL processed in the level of summarization.

USAGE NOTES: As the nonpaged pool is used, areas that were used and then freed are called holes. The size of the smallest hole is stored in SPRSMHL. When dynamic memory is required, if the size is less than SPRSMHL the memory manager allocates the space from that hole.

S P R S P F R E — Small Request Packets Free

SPRSPFRE contains the number of small request packets (SRPs) free.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMNSPF - Minimum Small Request Packets Free
SPRSPUSE - Small Request Packets in Use

ELEMENT DERIVATION: SPRSPFRE is set to the value of the MNR_POO$L_SRPCNT field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRSPFRE takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSPFRE contains the last value of SPRSPFRE processed in the level of summarization.

USAGE NOTES: The SYSGEN parameter SRPCOUNT determines how many SRPs are preallocated for use by processes on the system. When a process requests pool space that fits into an

SRP, one is allocated to the process. SPRSPFRE is the count of SRPs that are available for use.

S P R S P L I T Split Transfers

SPRSPLIT contains the count of I/O requests that were broken into two or more transfers.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRWTURN - Window Turns

ELEMENT DERIVATION: SPRSPLIT is set to the value of the MNR_IO$L_SPLTRANS field of the VMS MONITOR IO record.

RANGE OF VALUE: SPRSPLIT takes an integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSPLIT contains the sum of all values of SPRSPLIT processed in the level of summarization.

USAGE NOTES: When an I/O request is given to VMS and the cache buffers to receive it are too small, the request will be broken into as many requests as necessary to satisfy the I/O request. The requesting process will not be signaled that the I/O is complete until the last of the split transfers is done. Split transfers will lengthen the I/O response time to the process. If this value is high, consider increasing the size of the cache buffer.

S P R S P U S E Small Request Packets in Use

SPRSPUSE contains the number of small request packets (SRPs) in use.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMXSPU - Maximum Small Request Packets in Use
SPRSPFRE - Small Request Packets Free

ELEMENT DERIVATION: SPRSPUSE is set to the value of the MNR_POO$L_SRPINUSE field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRSPUSE takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSPUSE contains the last value of SPRSPUSE processed in the level of summarization.

USAGE NOTES: The SYSGEN parameter SRPCOUNT determines how many SRPs are preallocated for use by processes on the system. When a process requests pool space that fits into an SRP, one is allocated to the process. SPRSPUSE is the count of SRPs that are in use.

S P R S T A C T STA STATES Record Count (Type 1)

SPRSTACT contains a count of VMS MONITOR STATES records processed by the measurement interval.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS MONITOR STATES record is processed, SPRSTACT is set to 1.

RANGE OF VALUE: In the DETAIL timespan of the CA MICS VMS System Profile file, SPRSTACT is always 1. At higher levels of summarization, its value is the range from 1 to the total number of VMS MONITOR STATES records processed in the level of summarization.

SUMMARIZATION PROCESS: SPRSTACT contains the sum of all values of SPRSTACT processed in the level of summarization.

USAGE NOTES: SPRSTACT is used as the divisor for calculating averages and ratios in the level of summarization.

S P R S T A D R STA STATES Record Duration (Type 1)

SPRSTADR contains the number of seconds in the "reporting interval" of a given VMS MONITOR record. Reporting interval is the period of time during which a monitor gathers and reports statistics on this record.

TYPE: Common accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: SPRSTACT - STA STATES Record Count (Type 1)

```
          INTERVLS - Number of Recording Intervals
          STARTTS  - Start Time Stamp
          ENDTS    - End Time Stamp
          DEXDUR   - Requested Recording Interval Time

ELEMENT DERIVATION:  SPRSTADR contains the difference between
the time stamp of the current VMS MONITOR record and the time
stamp of the previously processed record of the same type.

RANGE OF VALUE:  The value of this element ranges from a
minimum of 0 to a maximum that is limited only by practical
bounds.

SUMMARIZATION PROCESS:  SPRSTADR contains the sum of all
occurrences of SPRSTADR that were included in the level of
summarization.

USAGE NOTES:  Note that SPRSTADR is uniquely calculated for
this VMS MONITOR record type before merging records together
into the DEMSPR file.


+-----------------+
| S P R S U S P   |  Suspended Wait, Inswapped
+-----------------+
SPRSUSP contains the count of processes that have been
suspended at their own request or at the request of some
other process.

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  PROSTATE - State
          SPRMXSPI - Maximum Suspended Wait, Inswapped
          SPRSUSPO - Suspended Wait, Outswapped

ELEMENT DERIVATION:  SPRSUSP is set to the value of the
MNR_STA$L_SUSP field from the VMS MONITOR STATES record.

RANGE OF VALUE:  SPRSUSP takes any integer value from 0 to
the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS:  SPRSUSP contains the last value of
SPRSUSP processed in the level of summarization.

USAGE NOTES:  The count in SPRSUSP does not affect VMS
performance to any great extent.  If the count is large and
the maximum number of processes allowed on the system is
reached, then processes that are suspended take up process
slots without doing any work at that time.  Suspended
processes usually will do useful work when they are taken off
suspension, so it is important to allow suspended processes
to remain on the system.


+-----------------+
| S P R S U S P O |  Suspended Wait, Outswapped
+-----------------+
SPRSUSPO contains the count of processes that have been
suspended at their own request or at the request of some
other process and now are outswapped.

TYPE:  Retained - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  PROSTATE - State
          SPRMXSPO - Maximum Suspended Wait, Outswapped
          SPRSUSP  - Suspended Wait, Inswapped

ELEMENT DERIVATION:  SPRSUSPO is set to the value of the
MNR_STA$L_SUSPO field from the VMS MONITOR STATES record.

RANGE OF VALUE:  SPRSUSPO takes any integer value from 0 to
the maximum number of processes allowed on the system.

SUMMARIZATION PROCESS:  SPRSUSPO contains the last value of
SPRSUSPO processed in the level of summarization.

USAGE NOTES:  The count in SPRSUSPO does not affect VMS
performance to any great extent.  If the count is large and
the maximum number of processes allowed on the system is
reached, then processes that are suspended take up process
slots without doing any work at that time.  Suspended
processes usually will do useful work when they are taken off
suspension, so it is important to allow suspended processes
to remain on the system.

Of all the processes to have in an outswapped state, the
processes counted in SPRSUSPO are the least costly.  Since
the process is indefinitely suspended, the extra time of
inswapping will have minimal effect on system performance.


+-----------------+
| S P R S Y S F L |  System Page Faults
+-----------------+
SPRSYSFL contains the number of system page faults that
occurred during DURATION (measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
```

bytes long.

SEE ALSO: SPRPSSYS - System Page Fault Rate

ELEMENT DERIVATION: SPRSYSFL is set to the value of the MNR_PAG$L_SYSFAULTS field of the VMS MONITOR PAGE record.

RANGE OF VALUE: SPRSYSFL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRSYSFL contains the sum of all values of SPRSYSFL processed in the level of summarization.

USAGE NOTES: System page faults are a potential bottleneck to all processes on the system. SPRSYSFL represents the number of times virtual addresses mapped to system space, but the required page was not available in the system working set. Since the system working set serves all processes on the system, the likelihood of a page being needed by another process immediately is low. This means that when one process causes a page to be read in, a page for another process was written out. The other process likely will need that page again when it processes, so it will also fault. This chain reaction causes system space thrashing and will quickly degrade the performance of the system.

System working set size should be increased to the point where SPRSYSFL is consistently small. The SYSGEN parameter that controls this is SYSMWCNT. SYSMWCNT should not be too large, however, because that space is taken away from the memory available for process working sets, which also could cause paging and swapping bottlenecks.

S P R S Y T C T — SYT SYSTEM Record Count (Type 17)

SPRSYTCT contains a count of VMS MONITOR SYSTEM records processed by the CA MICS VAX/VMS Analyzer.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS MONITOR SYSTEM record is processed, SPRSYTCT is set to 1.

RANGE OF VALUE: In the DETAIL timespan of the CA MICS VMS System Profile file, SPRSYTCT is always 0 or 1. At higher levels of summarization, its value ranges from 0 to the total number of VMS MONITOR SYSTEM records processed in the level of summarization.

SUMMARIZATION PROCESS: SPRSYTCT contains the sum of all values of SPRSYTCT processed in the level of summarization.

USAGE NOTES: SPRSYTCT is used as the divisor for calculating averages and ratios in the level of summarization.

S P R S Y T D R — SYT SYSTEM Record Duration (Type 17)

SPRSYTDR contains the number of seconds in the "reporting interval" of a given VMS MONITOR record. Reporting interval is the period of time during which a monitor gathers and reports statistics on this record.

TYPE: Common accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: SPRSYTCT - SYT SYSTEM Record Count (Type 17)
INTERVLS - Number of Recording Intervals
STARTTS - Start Time Stamp
ENDTS - End Time Stamp
DEXDUR - Requested Recording Interval Time

ELEMENT DERIVATION: SPRSYTDR contains the difference between the time stamp of the current VMS MONITOR record and the time stamp of the previously processed record of the same type.

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SPRSYTDR contains the sum of all occurrences of SPRSYTDR that were included in the level of summarization.

USAGE NOTES: Note that SPRSYTDR is uniquely calculated for this VMS MONITOR record type before merging records together into the DEMSPR file.

S P R T R N C T — TRN TRANS Record Count (Type 22)

SPRTRNCT contains a count of VMS MONITOR TRANSACTION records processed by the CA MICS VAX/VMS Analyzer.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: When a VMS MONITOR TRANSACTION record is processed, SPRTRNCT is set to 1.

RANGE OF VALUE: In the DETAIL timespan of the CA MICS VMS System Profile file, SPRTRNCT is always 0 or 1. At higher levels of summarization, its value ranges from 0 to the total number of VMS MONITOR TRANSACTION records processed in the level of summarization.

SUMMARIZATION PROCESS: SPRTRNCT contains the sum of all values of SPRTRNCT processed in the level of summarization.

USAGE NOTES: SPRTRNCT is used as the divisor for calculating averages and ratios in the level of summarization.

S P R T R N D R | TRN TRANS Record Duration (Type 22)

SPRTRNDR contains the number of seconds in the "reporting interval" of a given VMS MONITOR record. Reporting interval is the period of time during which a monitor gathers and reports statistics on this record.

TYPE: Common accumulated data element - Numeric, measurement unit is integer.

SEE ALSO: SPRTRNCT - TRN TRANS Record Count (Type 22)
INTERVLS - Number of Recording Intervals
STARTTS - Start Time Stamp
ENDTS - End Time Stamp
DEXDUR - Requested Recording Interval Time

ELEMENT DERIVATION: SPRTRNDR contains the difference between the time stamp of the current VMS MONITOR record and the time stamp of the previously processed record of the same type.

RANGE OF VALUE: SPRTRNDR ranges from a minimum of 0 to a maximum limited only by practical bounds.

SUMMARIZATION PROCESS: SPRTRNDR contains the sum of all occurrences of SPRTRNDR that were included in the level of summarization.

USAGE NOTES: Note that SPRTRNDR is uniquely calculated for this VMS MONITOR record type before merging records together into the DEMSPR file.

S P R T R N G E | DEXTS Range From Input Records

Range, or maximum difference, of DEXTS values from the source records used to construct the DEMSPR file observation.

TYPE: Maximum data element -- NUMERIC, measurement unit is INTEGER.

SEE ALSO: DEXTS - Monitor Time Stamp

ELEMENT DERIVATION: This element is computed as the maximum of ABS(DEXTS-ENDTS) for each set of records that build an observation in the DEMSPR file.

RANGE OF VALUE: The value of this element ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SPRTRNGE contains the sum of all occurrences of SPRTRNGE that were included in the level of summarization.

USAGE NOTES: This element is not normally used, unless you need to examine the method by which DEMSPR file is constructed. SPRTRNGE is provided to help determine if the difference between ENDTS in the DEMSPR file and DEXTS in the various source records for the file is significant.

S P R T U R N | Window Turns

SPRTURN contains the number of I/Os that were requested but would not fit in the I/O buffer window.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRPSTRN - Window Turn Rate
SPRWTURN - Window Turns

ELEMENT DERIVATION: SPRTURN is set to the value of the MNR_FCP$L_ field of the VMS MONITOR FCP record.

RANGE OF VALUE: SPRTURN takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRTURN contains the sum of all values of SPRTURN processed in the level of summarization.

USAGE NOTES: Window turns occur when the file map is not completely in the system buffers. To access all portions of the file system, multiple pages of the file map need to be read in to find the physical locations of the portions needed. On systems where there are many small files created and deleted, the file system gradually becomes fragmented into small pieces over time. When large files are allocated on a fragmented file system, they are broken up into several noncontiguous pieces. When the system performs I/O to large fragmented files, system performance can be negatively affected by the amount of I/O time and CPU processing necessary to perform these window turns. Usually when SPRTURN is large, the system is thrashing on the file map and performance is severely degraded. Note that on versions of VMS prior to version 5.0, SPRWTURN is used to get the same information.

S P R U N C O N — Nonpaged Pool Unused Contiguous Space

SPRUNCON contains the number of unused contiguous space in the dynamic nonpaged pool.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRMNUCN - Minimum Unused Contiguous Space

ELEMENT DERIVATION: SPRUNCON is set to the value of the MNR_POO$L_HOLECNT field of the VMS MONITOR POOL record.

RANGE OF VALUE: SPRUNCON takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRUNCON contains the last value of SPRUNCON processed in the level of summarization.

USAGE NOTES: As the dynamic pool becomes fragmented, SPRUNCON keeps track of the amount of free contiguous space. If SPRUNCON is growing, the pool badly fragment and degrade performance.

S P R U N I — Unlocks, Incoming

SPRUNI contains the count of incoming dequeues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRUNO - Unlocks, Outgoing
SPRUNL - Unlocks, Local

ELEMENT DERIVATION: SPRUNI is set to the value of the MNR_DLO$L_DEQIN field of the VMS MONITOR DLOCK record.

RANGE OF VALUE: SPRUNI takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRUNI contains the sum of all values of SPRUNI processed in the level of summarization.

USAGE NOTES: SPRUNI counts the number of locks from remote nodes that were released (dequeued).

S P R U N L — Unlocks, Local

SPRUNL contains the count of local dequeues.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: SPRUNO - Unlocks, Outgoing
SPRUNI - Unlocks, Incoming

ELEMENT DERIVATION: SPRUNL is set to the value of the MNR_DLO$L_DEQLOC field of the VMS MONITOR DLOCK record.

RANGE OF VALUE: SPRUNL takes any integer value from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: SPRUNL contains the sum of all values of SPRUNL processed in the level of summarization.

USAGE NOTES: SPRUNL counts the number of locks from this node that were released (dequeued).

```
+-----------------+
| S P R U N O     |  Unlocks, Outgoing
+-----------------+
SPRUNO contains the count of outgoing dequeues.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRUNI - Unlocks, Incoming
           SPRUNL - Unlocks, Local

ELEMENT DERIVATION:  SPRUNO is set to the value of the
MNR_DLO$L_DEQOUT field of the VMS MONITOR DLOCK record.

RANGE OF VALUE:  SPRUNO takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRUNO contains the sum of all values
of SPRUNO processed in the level of summarization.

USAGE NOTES:  SPRUNO counts the number of locks from this
node that were released (dequeued) from remote nodes.


+-----------------+
| S P R V L O K   |  Volume Lock Waits
+-----------------+
SPRVLOK contains the number of times the file system
processor was waiting on a volume lock to be freed during
DURATION (measurement interval).

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  Not applicable

ELEMENT DERIVATION:  SPRVLOK is set to the value of the
MNR_FCP$L_VOLWAIT field of the VMS MONITOR FCP record.

RANGE OF VALUE:  SPRVLOK takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRVLOK contains the sum of all
values of SPRVLOK processed in the level of summarization.

USAGE NOTES:  When file system operations involve changing
the disk volume-wide information, volume access must be
serialized to prevent collisions by competing processes.
SPRVLOK is the number of times processes had to wait for
their turns.  If SPRVLOK is large, I/O response could be
elongated because of contention for volume access.


+-----------------+
| S P R W I P F L |  Write In Progress Page Faults
+-----------------+
SPRWIPFL contains the number of page faults in DURATION
(measurement interval) that were satisfied by pages that were
in the process of being written out.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRMPGFL - Modified List Page Faults
           SPRPSWIP - Write In Progress Page Fault Rate

ELEMENT DERIVATION:  SPRWIPFL is set to the value of the
MNR_PAG$L_WRTINPROG field of the VMS MONITOR PAGE record.

RANGE OF VALUE:  SPRWIPFL takes any integer value from 0 to
2,147,483,647.

SUMMARIZATION PROCESS:  SPRWIPFL contains the sum of all
values of SPRWIPFL processed in the level of summarization.

USAGE NOTES:  SPRWIPFL is typically 0 or close to zero. This
is an unusual class of page faults; the pages were marked as
removable, but as the memory manager was writing them out,
the processes accessed them again.  If SPRWIPFL becomes
large, it is probably a fluke.  If it is consistent, there is
likely a severe problem with the I/O subsystem.


+-----------------+
| S P R W T U R N |  Window Turns
+-----------------+
SPRWTURN contains the count of I/O requests that would not
fit in the I/O buffer window.

TYPE:  Accumulated - Numeric, measurement unit is integer, 4
bytes long.

SEE ALSO:  SPRTURN - Window Turns

ELEMENT DERIVATION:  SPRWTURN is set to the value of the
MNR_IO$L_WTURNS field of the VMS MONITOR IO record.

RANGE OF VALUE:  SPRWTURN takes an integer value from 0 to
2,147,483,647.
```

SUMMARIZATION PROCESS: SPRWTURN contains the sum of all values of SPRWTURN processed in the level of summarization.

USAGE NOTES: SPRWTURN is only in I/O class records prior to VMS version 5.0. In VMS version 5.0 and above systems, this information is located in SPRTURN of the VMS MONITOR FCP record. Window turns occur when the file map is not completely in the system buffers. To access all portions of the file system, multiple pages of the file map need to be read in to find the physical locations of the portions needed. On systems where there are many small files that are created and deleted, the file system gradually becomes fragmented into small pieces over time. When large files are allocated on a fragmented file system, they are broken up into several noncontiguous pieces. When the system performs I/O to large fragmented files, system performance can be negatively affected by the amount of I/O time and CPU processing necessary to perform these window turns. Usually when SPRWTURN is large, the system is thrashing on the file map and performance is severely degraded.

B.4 - VAX/VMS System Usage Information (DES) Area

The sections that follow describe the VAX/VMS System Usage information area (DES). The statistics for these files are obtained from the CA MICS System Usage (DEXSUS) collector routine.

1 - File Relationships and Common Data Elements
2 - The VMS Disk Device (DESDKD) File
3 - The VMS Disk Quota (DESDKQ) File
4 - The VMS Disk Usage (DESDKU) File
5 - The VMS Process Activity (DESPRX) File
6 - The VMS System Status (DESSYU) File

B.4.1 - File Relationships and Common Data Elements

Not available at this time.

B.4.2 - The VMS Disk Device (DESDKD) File

FILE ABSTRACT: VMS Disk Device File	
CONTENT	Identifies disk device characteristics and overall space usage for each active VMS disk.
USE	To report total space and available space for storage management and capacity planning.
SOURCE	VMS ANALYZE/DISK_STRUCTURE Utility identification (type 1) record as documented in the VMS Analyze/Disk_Structure Utility Manual, and VMS SHOW DEVICE/FULL command output.
TABLES	N/A

Generation Date: Thu, Jul 18, 2019

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXDEVIC	DEXVOLNM	DKDDEVTY	DKDVSNAM
	YEAR	MONTH	DAY	ENDTS	
DAYS	SYSID	DEXDEVIC	DEXVOLNM	DKDDEVTY	DKDVSNAM
	YEAR	MONTH	DAY		
WEEKS	SYSID	DEXDEVIC	DEXVOLNM	DKDDEVTY	DKDVSNAM
	YEAR	WEEK			
MONTHS	SYSID	DEXDEVIC	DEXVOLNM	DKDDEVTY	DKDVSNAM
	YEAR	MONTH			
YEARS	N/A				

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XDWM..E	DEXDEVIC	- Device Name
XDWM..E	DEXVOLNM	- Volume Name
XDWM..E	DKDDEVTY	- Device Type
XDWM..E	DKDVSNAM	- Volume Set Name
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
Common Data Elements		
XD....E	DAYNAME	- Name of Day of Week
X.....E	DEXNODE	- Nodename

```
 XDWM..E  DEXSUSTB - DEXSUS System Boot Time Stamp
 XDWM..E  DEXSUSTS - DEXSUS Collect Time Stamp
 X.....E  DEXVOLSR - Volume Serial Number
 XDWM..E  ENDTS    - End Time Stamp
 XDW...E  HOUR     - Hour of Day
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  ZONE     - Time Zone
```

Retained Data Elements

```
 XDWM..E  DKDBLKSF - Device Free Blocks
 XDWM..E  DKDBLKST - Device Total Blocks
 XDWM..E  DKDERRCT - Device Error Count
 XDWM..E  DKDVOFMT - Volume Format Type
 XDWM..E  DKDVOWNM - Volume Owner Name
```

```
+-----------------+
| D K D B L K S F |  Device Free Blocks
+-----------------+
```

DKDBLKSF contains the count of free blocks of space (512 bytes) on the device.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: DKDBLKST - Device Total Blocks

ELEMENT DERIVATION: DKDBLKSF is set to the "Free blocks" value reported by SHOW DEV/FULL for the device. This information is collected in the DEXSUS DISK DEVICE measurement records set.

RANGE OF VALUE: DKDBLKSF takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds for the specific device type.

SUMMARIZATION PROCESS: DKDBLKSF contains the last value of DKDBLKSF processed in the level of summarization.

USAGE NOTES: VMS Disk blocks are 512 bytes in size.

```
+-----------------+
| D K D B L K S T |  Device Total Blocks
+-----------------+
```

DKDBLKST contains the count of total blocks of space (512 bytes) on the device.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: DKDBLKSF - Device Free Blocks

ELEMENT DERIVATION: DKDBLKST is set to the "Total blocks" value reported by SHOW DEV/FULL for the device. This information is collected in the DEXSUS DISK DEVICE measurement records set.

RANGE OF VALUE: DKDBLKST takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds for the specific device type.

SUMMARIZATION PROCESS: DKDBLKST contains the last value of DKDBLKST processed in the level of summarization.

USAGE NOTES: Disk Blocks are 512 bytes in size.

```
+-----------------+
| D K D D E V T Y |  Device Type
+-----------------+
```

DKDDEVTY contains the device type designation for the disk device.

TYPE: Retained - Character, length 8.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DKDDEVTY is set to the "device type" value reported by SHOW DEV/FULL for the device. This information is collected in the DEXSUS DISK DEVICE measurement records set.

RANGE OF VALUE: DKDDEVTY can contain any combination of 1 to 8 alphanumeric characters.

SUMMARIZATION PROCESS: DKDDEVTY is not summarized. DKDDEVTY is used to determine a unique key for records within a file based on the time-span, so other elements in the file can be summarized.

USAGE NOTES: Example of a typical value: RA81.

```
+-----------------+
| D K D E R R C T |  Device Error Count
+-----------------+
```

DKDERRCT contains the most recent error count reported for the disk device.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DKDERRCT is set to the "Error count" value reported by SHOW DEV/FULL for the device. This information is collected in the DEXSUS DISK DEVICE measurement records set.

RANGE OF VALUE: DKDERRCT takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DKDERRCT contains the last value of DKDERRCT processed in the level of summarization.

USAGE NOTES: None

D K D V O F M T — Volume Format Type

DKDVOFMT contains the volume format type for the disk device.

TYPE: Retained - Character, length 12.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DEXVOFMT is set to the value of the USG$T_FORMAT field of the Analyze/Disk Identification Record (type 1).

RANGE OF VALUE: DKDVOFMT can contain any combination of 1 to 12 alphanumeric characters.

SUMMARIZATION PROCESS: DKDVOFMT contains the last value of DKDVOFMT processed in the level of summarization.

USAGE NOTES: The value of this field can be interpreted as follows:

DECFILE11A = Files-11 Structure Level 1 volume
DECFILE11B = Files-11 Structure Level 2 volume

D K D V O W N M — Volume Owner Name

DKDVOWNM contains the volume owner name of the disk device.

TYPE: Retained - Character, length 12.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DKDVOWNM is set to the value of the USG$T_OWNERNAME field of the Analyze/Disk Identification Record (type 1).

RANGE OF VALUE: DKDVOWNM can contain any combination of 1 to 12 alphanumeric characters.

SUMMARIZATION PROCESS: DKDVOWNM contains the last value of DKDVOWNM processed in the level of summarization.

USAGE NOTES: None

D K D V S N A M — Volume Set Name

DKDVSNAM contains the volume set name if the volume is part of a volume set.

TYPE: Retained - Character, length 12.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DKDVSNAM is set to the value of the USG$T_STRUCNAM field of the Analyze/Disk Identification Record (type 1).

RANGE OF VALUE: DKDVSNAM can contain any combination of 1 to 12 alphanumeric characters.

SUMMARIZATION PROCESS: DKDVSNAM is not summarized. DKDVSNAM is used to determine a unique key for records within a file based on the time-span, so other elements in the file can be summarized.

USAGE NOTES: For a Files-11 Structure Level 1 volume, or for a Files-11 Structure Level 2 volume that is not part of a volume set, this element contains blanks.

B.4.3 - The VMS Disk Quota (DESDKQ) File

FILE ABSTRACT: VMS Disk Quota File	
CONTENT	Quantifies disk space usage by Diskquota-displayed User ID Codes (UIC's) for each active quota-enabled VMS disk.
USE	To report quota usage, permanent quotas and overdraft limits, and storage occupancy for storage management and capacity planning.
SOURCE	VMS SYSMAN Utility, DISKQUOTA SHOW output

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXDEVIC	DKQUIC	YEAR	MONTH
	DAY	ENDTS			
DAYS	SYSID	DEXDEVIC	DKQUIC	YEAR	MONTH
	DAY				
WEEKS	SYSID	DEXDEVIC	DKQUIC	YEAR	WEEK
MONTHS	SYSID	DEXDEVIC	DKQUIC	YEAR	MONTH
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
	Sequence/Summary Data Elements	
XD....E	DAY	- Day of Month
XDWM..E	DEXDEVIC	- Device Name
XDWM..E	DKQUIC	- Disk Quota UIC
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
	Common Data Elements	
XD....E	DAYNAME	- Name of Day of Week
X.....E	DEXNODE	- Nodename
XDWM..E	DEXSUSTB	- DEXSUS System Boot Time Stamp
XDWM..E	DEXSUSTS	- DEXSUS Collect Time Stamp
XDWM..E	DURATION	- Recording Interval Time
XDWM..E	ENDTS	- End Time Stamp
XDW...E	HOUR	- Hour of Day

```
 X.....E  ORGSYSID - Originating System Identification
 XDWM..E  STARTTS  - Start Time Stamp
 XDWM..E  ZONE     - Time Zone

Retained Data Elements

 X.....E  DKQOVLIM - Disk Quota Overdraft Limit (Blocks)
 X.....E  DKQPQUOT - Disk Quota Permanent Quota (Blocks)

Accumulated Data Elements

 XDWM..E  DKQMBHUS - Disk Quota Usage Megabyte (2**20) Hrs
 X.....E  DKQUSAGE - Disk Quota Usage (Blocks)
```

D K Q M B H U S — Disk Quota Usage Megabyte (2**20) Hrs

DKQMBHUS contains disk space occupancy for the specified UIC and device as measured by Disk Quota.

TYPE: Accumulated - Numeric, measurement unit is real number.

SEE ALSO: DKQUSAGE - Disk Quota Usage (Blocks)

ELEMENT DERIVATION: DKQMBHUS is calculated as follows:

```
DKQMBHUS = (DURATION/(60*60)) * (DKQUSAGE*512) / (2**20);
```

RANGE OF VALUE: DKQMBHUS takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DKQMBHUS contains the sum of all values of DKQMBHUS processed in the level of summarization.

USAGE NOTES: DKQMBHUS is based on DKQUSAGE measurement from the DISKQUOTA SHOW command. This measurement source is not typically accurate enough to provide the basis for disk usage accounting. However, this source can be kept relatively accurate for overall space management and quota management, if disk quota enable status is maintained for each measured disk and disk quota rebuild is run as required. The Analyze/Disk utility reports discrepancies between quota usage measures and observed usage on each disk analyzed.

DURATION is calculated in the DETAIL level input processing as time elapsed between the time stamp of the current DEXSUS data collection event being input and the previous reading for the same device, and UIC from either the input data file or the previous day's DETAIL data. If no such reading is available for the UIC, the elapsed time since the last DEXSUS collection event is assumed. If this elapsed time can not be determined, then the default duration specified in the DEXOPS parameter member is assumed.

D K Q O V L I M — Disk Quota Overdraft Limit (Blocks)

DKQOVLIM contains the overdraft limit value for the specified UIC and device as reported by Disk Quota.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DKQOVLIM is set to the "Overdraft Limit" value reported by DISKQUOTA SHOW for the UIC and device. This information is collected in the DEXSUS DISK QUOTA measurement records set.

RANGE OF VALUE: DKQOVLIM takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DKQOVLIM contains the last value of DKQOVLIM processed in the level of summarization.

USAGE NOTES: None

D K Q P Q U O T — Disk Quota Permanent Quota (Blocks)

DKQPQUOT contains the permanent quota value for the specified UIC and device as reported by Disk Quota.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DKQPQUOT is set to the "Permanent Quota" value reported by DISKQUOTA SHOW for the UIC and device. This information is collected in the DEXSUS DISK QUOTA measurement records set.

RANGE OF VALUE: DKQPQUOT takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DKQPQUOT contains the last value of DKQPQUOT processed in the level of summarization.

USAGE NOTES: None

| D K Q U I C | Disk Quota UIC

DKQUIC contains the User ID Code (Group and Member) from the quota statistics records reported by Disk Quota.

TYPE: Retained - Character, length 25.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DKQUIC is set to the "UIC" value reported by DISKQUOTA SHOW for each disk quota statistics record. This information is collected in the DEXSUS DISK QUOTA measurement records set.

RANGE OF VALUE: DKQUIC can contain any combination of 1 to 25 alphanumeric characters.

SUMMARIZATION PROCESS: DKQUIC is not summarized. DKQUIC is used to determine a unique key for records within a file based on the timespan, so other elements in the file can be summarized.

USAGE NOTES: Disk Quota UIC values are stored as recorded. This data is typically reported as a combination of numeric and character UIC Group and Member codes. There is no direct method to map these codes to the UIC Group and Member codes in other CA MICS VAX/VMS Analyzer files.

| D K Q U S A G E | Disk Quota Usage (Blocks)

DKQUSAGE contains the value for disk usage in blocks of 512 bytes for the specified UIC and device as reported by Disk Quota.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: DKQMBHUS - Disk Quota Usage Megabyte (2**20) Hrs

ELEMENT DERIVATION: DKQUSAGE is set to the "Usage" value reported by DISKQUOTA SHOW for the UIC and device. This information is collected in the DEXSUS DISK QUOTA measurement records set.

RANGE OF VALUE: DKQUSAGE takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DKQUSAGE contains the sum total of DKQUSAGE processed in the level of summarization.

USAGE NOTES: DKQMBHUS is based on DKQUSAGE measurement from the DISKQUOTA SHOW command. This measurement source is not typically accurate enough to provide the basis for disk usage accounting. However, this source can be kept relatively accurate for overall space management and quota management, if disk quota enable status is maintained for each measured disk and disk quota rebuild is run as required. The Analyze/Disk utility reports discrepancies between quota usage measures and observed usage on each disk analyzed.

B.4.4 - The VMS Disk Usage (DESDKU) File

FILE ABSTRACT: VMS Disk Usage File	
CONTENT	Quantifies disk usage by account code levels, directory top level, and by individual VMS directories and files for each active VMS disk.
USE	To report space allocated and used, and storage occupancy for usage accounting, storage management, and capacity planning.
SOURCE	VMS ANALYZE/DISK_STRUCTURE Utility file (type 2) record as documented in the VMS Analyze/Disk_Structure Utility Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	DEXDEVIC	DKUDIRTL	DKUDIR	DKUFILE
	YEAR	MONTH	DAY	ENDTS	DEXACT1
	DEXACT2	DEXACT3			
DAYS	SYSID	DEXDEVIC	DKUDIRTL	YEAR	MONTH
	DAY	DEXACT1	DEXACT2	DEXACT3	
WEEKS	N/A				
MONTHS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
	MONTH				
YEARS	SYSID	DEXACT1	DEXACT2	DEXACT3	YEAR
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XD.MY.E	DEXACT1	- DIVISION
XD.MY.E	DEXACT2	- UIC GROUP
XD.MY.E	DEXACT3	- USER NAME
XD....E	DEXDEVIC	- Device Name
X.....E	DKUDIR	- Directory Name
XD....E	DKUDIRTL	- Directory Top Level
X.....E	DKUFILE	- File Specification
XD.M..E	MONTH	- Month of Year
XD.MY.E	SYSID	- System Identifier
XD....E	WEEK	- Week of Year
XD.MY.E	YEAR	- Year of Century
Common Data Elements		

```
 XD....E  DAYNAME  - Name of Day of Week
 X.....E  DEXGRP   - User Identification Code Group
 X.....E  DEXMBR   - User Identification Code Member
 X.....   DEXNODE  - Nodename
 XD.MY.   DEXSUSTB - DEXSUS System Boot Time Stamp
 XD.MY.   DEXSUSTS - DEXSUS Collect Time Stamp
 X.....E  DEXVOLNM - Volume Name
 X.....E  DEXVOLSR - Volume Serial Number
 XD.MY.E  DURATION - Recording Interval Time
 XD.MY.E  ENDTS    - End Time Stamp
 XD....E  HOUR     - Hour of Day
 X.....E  ORGSYSID - Originating System Identification
 XD.MY.E  STARTTS  - Start Time Stamp
 XD.MY.E  ZONE     - Time Zone
```

Accumulated Data Elements

```
 X.....E  DKUBLKSA - Number of Blocks (512) Allocated
 X.....   DKUBLKST - Number of Blocks (512) Total
 X.....E  DKUBLKSU - Number of Blocks (512) Used
 XD.MY.E  DKUCNT   - File Count
 XD.MY.E  DKUCOST  - Disk Usage Charges
 XD.MY.E  DKUMBHA  - Megabyte (2**20) Hours Allocated
 XD.MY.E  DKUMBHU  - Megabyte (2**20) Hours Used
```

D K U B L K S A — Number of Blocks (512) Allocated

DKUBLKSA contains the number of blocks allocated for data storage on the disk device, for the specified file or collection of files.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: DKUBLKSU - Number of Blocks (512) Used
DKUBLKST - Number of Blocks (512) Total

ELEMENT DERIVATION: DKUBLKSA is set to the value of DKUBLKST minus one in the DETAIL time-span.

RANGE OF VALUE: DKUBLKSA takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DKUBLKSA contains the sum total of DKUBLKSA processed in the level of summarization.

USAGE NOTES: DKUBLKSA represents the number of blocks allocated for the file, excluding the block reserved for the file header information. This value should agree with the blocks allocated value reported by the VMS Directory command. DKUBLKSA is calculated as one block less than the DKUBLKST value reported from Analyze/Disk for number of blocks allocated to the file.

D K U B L K S T — Number of Blocks (512) Total

DKUBLKST contains the total number of blocks allocated, including data and file header space, on the disk device, for the specified file or collection of files.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: DKUBLKSU - Number of Blocks (512) Used
DKUBLKSA - Number of Blocks (512) Allocated

ELEMENT DERIVATION: DKUBLKST is obtained from the Analyze/Disk Usage File Record (type 2), from USG$L_ALLOCATED.

RANGE OF VALUE: DKUBLKST takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DKUBLKST contains the sum total of DKUBLKST processed in the level of summarization.

USAGE NOTES: DKUBLKST represents the number of blocks allocated for the file, including the block reserved for the file header information. DKUBLKSA is calculated as one block less than the DKUBLKST value reported from Analyze/Disk for number of blocks allocated to the file.

D K U B L K S U — Number of Blocks (512) Used

DKUBLKSU contains the number of blocks actually in use for data storage on the disk device, for the specified file or collection of files.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: DKUBLKST - Number of Blocks (512) Total
DKUBLKSA - Number of Blocks (512) Allocated

ELEMENT DERIVATION: DKUBLKSU is obtained from the Analyze/Disk Usage File Record (type 2), from USG$L_USED.

RANGE OF VALUE: DKUBLKSU takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DKUBLKSU contains the sum total of DKUBLKSU processed in the level of summarization.

USAGE NOTES: None

D K U C N T — File Count

DKUCNT contains a count of the number of file records the measurement represents.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: DKUCNT is set to a value of 1 in each of the detail disk usage file records.

RANGE OF VALUE: In the DETAIL time-span, DKUCNT's value is always 1. At higher levels of summarization, its value ranges from a minimum of 1 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DKUCNT contains the sum of all values of DKUCNT processed in the level of summarization.

USAGE NOTES: This element provides a sample count for averaging disk usage measurements and for counting the number of files per directory, disk device, or other groupings.

D K U C O S T — Disk Usage Charges

DKUCOST contains the disk storage costs for each VMS file. The value of DKUCOST is zero unless a user computation is performed or the file containing this element is selected for billing processing by CA MICS Accounting and Chargeback.

TYPE: Accumulated - Numeric, measurement unit is real number.

SEE ALSO: Not applicable

ELEMENT DERIVATION: In the DETAIL timespan, DKUCOST is set by the Accounting and Chargeback Interface.

RANGE OF VALUE: DKUCOST is determined by the charging algorithms in effect at your installation.

SUMMARIZATION PROCESS: DKUCOST contains the sum of all values of DKUCOST processed in the level of summarization.

USAGE NOTES: None

D K U D I R — Directory Name

DKUDIR contains the directory name portion of the VMS file specification.

TYPE: Retained - Character, length 40.

SEE ALSO: DKUDIRTL - Directory Top Level
DKUFILE - File Specification

ELEMENT DERIVATION: DKUDIR is obtained from the Analyze/Disk Usage File Record (type 2), from that portion of USG$T_FILESPEC representing the directory name. This string is delineated by the value of USG$W_DIR_LEN in the same record. The brackets are removed from the string before storage in DKUDIR.

RANGE OF VALUE: DKUDIR can contain any combination of 1 to 40 alphanumeric characters.

SUMMARIZATION PROCESS: DKUDIR is not summarized. DKUDIR is used to determine a unique key for records within a file based on the time-span, so other elements in the file can be summarized.

USAGE NOTES: VMS files are named in logical groupings called directories. DKUDIR can be used to help identify the owner and use of the files in such a grouping. Space usage by directory can also be tracked to help manage storage growth.

D K U D I R T L — Directory Top Level

DKUDIRTL contains the first level of the directory name of the VMS file, up to 12 characters in length.

TYPE: Retained - Character, length 12.

SEE ALSO: DKUDIR - Directory Name
DKUFILE - File Specification

ELEMENT DERIVATION: DKUDIRTL is set to first level of the directory name in DKUDIR for each VMS file. The level is delimited in the name string by a period, or by a maximum length of 12 characters.

RANGE OF VALUE: DKUDIRTL can contain any combination of 1 to 12 alphanumeric characters.

SUMMARIZATION PROCESS: DKUDIRTL is not summarized. DKUDIRTL is used to determine a unique key for records within a file based on the time-span, so other elements in the file can be summarized.

USAGE NOTES: VMS files are named in logical groupings called directories, which are formed in a tree structure. DKUDIRTL can be used to provide a high level of grouping, beyond potentially numerous variations of DKUDIR, to help identify the owner and use of the files in such a grouping. Space usage by directory group can also be tracked to help manage storage growth.

D K U F I L E — File Specification

DKUFILE contains the file name, type, and version of the VMS file specification.

TYPE: Retained - Character, length 40.

SEE ALSO: DKUDIR - Directory Name
DKUDIRTL - Directory Top Level

ELEMENT DERIVATION: DKUFILE is obtained from the Analyze/Disk Usage File Record (type 2), from the portion of USG$T_FILESPEC that represents the file name, type, and version. This string is delineated by the value of USG$W_DIR_LEN and USG$W_SPEC_LEN in the same record. The remaining portion of the full file specification following the directory name is assigned to DKUFILE.

RANGE OF VALUE: DKUFILE can contain any combination of 1 to 40 alphanumeric characters.

SUMMARIZATION PROCESS: DKUFILE is not summarized. DKUFILE is used to determine a unique key for records within a file based on the time-span, so other elements in the file can be summarized.

USAGE NOTES: DKUFILE is stored only at the DETAIL level time-span due to the space requirements for storing each file name instance at higher levels of summarization. DKUDIR and DKUDIRTL elements can be useful over time for space management at the user level.

VMS file names contain a version number for each level of file modification. The DKUFILE value includes the version number portion of the file specification, in the form of a trailing semicolon and numeric value string. This fact should be considered when processing VMS file names.

D K U M B H A — Megabyte (2**20) Hours Allocated

DKUMBHA contains the disk space occupancy of allocated blocks, measured in megabyte hours.

TYPE: Accumulated - Numeric, measurement unit is real number.

SEE ALSO: DKUMBHU - Megabyte (2**20) Hours Used
DKUBLKSU - Number of Blocks (512) Used
DKUBLKST - Number of Blocks (512) Total
DKUBLKSA - Number of Blocks (512) Allocated

ELEMENT DERIVATION: DKUMBHA is calculated as:

```
DKUMBHA = (DURATION/(60*60)) * (DKUBLKSA*512) / (2**20);
```

RANGE OF VALUE: DKUMBHA's value ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: DKUMBHA contains the sum of all values of DKUMBHA processed in the level of summarization.

USAGE NOTES: Disk space occupancy calculation requires both a space measurement and an elapsed time assumption to compute a value. DURATION is calculated in the detail level input processing as time elapsed between the time stamp of the current DEXSUS data collection event being input and the previous reading for the same file from either the input data file or the previous day's DETAIL data. If no such reading is available for the file, the elapsed time since the last DEXSUS collection event is assumed. If this elapsed time cannot be determined, then the default duration specified in the DEXOPS parameter member is assumed.

```
+-----------------+
| D K U M B H U   |  Megabyte (2**20) Hours Used
+-----------------+
DKUMBHU contains the disk space occupancy of blocks used,
measured in megabyte hours.

TYPE:  Accumulated - Numeric, measurement unit is real
number.

SEE ALSO:  DKUMBHU  - Megabyte (2**20) Hours Used
           DKUBLKSU - Number of Blocks (512) Used
           DKUBLKST - Number of Blocks (512) Total
           DKUBLKSA - Number of Blocks (512) Allocated

ELEMENT DERIVATION:  DKUMBHU is calculated as:

  DKUMBHU = (DURATION/(60*60)) * (DKUBLKSU*512) / (2**20);

RANGE OF VALUE:  DKUMBHU takes any numeric value in the range
from a minimum of 0 to a maximum that is limited only by
practical bounds.

SUMMARIZATION PROCESS:  DKUMBHU contains the sum of all
values of DKUMBHU processed in the level of summarization.

USAGE NOTES:  Disk space occupancy calculation requires both
a space measurement and an elapsed time assumption to compute
a value.  DURATION is calculated in the detail level input
processing as time elapsed between the time stamp of the
current DEXSUS data collection event being input and the
previous reading for the same file from either the input data
file or the previous day's DETAIL data.  If no such reading
is available for the file, the elapsed time since the last
DEXSUS collection event is assumed.  If this elapsed time
cannot be determined, then the default duration specified in
the DEXOPS parameter member is assumed.
```

B.4.5 - The VMS Process Activity (DESPRX) File

FILE ABSTRACT: VMS Process Activity File	
CONTENT	Quantifies the resource usage of each process during the monitoring interval, with added identification information from accounting.
USE	To report resource usage of each process for performance tuning and capacity planning.
SOURCE	VMS MONITOR record type 0 as documented in the VMS Monitor Reference Manual, plus VMS ACCOUNTING record type 01 as documented in the VMS Accounting Reference Manual.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	PRXTYPE	PRXPNAME	DEXUSER	DEXACCNT
	DEXGRP	DEXMBR	DEXTERM	YEAR	MONTH
	DAY	HOUR	ENDTS		
DAYS	SYSID	PRXTYPE	PRXPNAME	DEXUSER	DEXACCNT
	DEXGRP	DEXMBR	DEXTERM	YEAR	MONTH
	DAY	HOUR			
WEEKS	SYSID	PRXTYPE	PRXPNAME	DEXUSER	DEXACCNT
	DEXGRP	DEXMBR	DEXTERM	YEAR	WEEK
	ZONE	HOUR			
MONTHS	SYSID	PRXTYPE	PRXPNAME	DEXUSER	DEXACCNT
	DEXGRP	DEXMBR	DEXTERM	YEAR	MONTH
	ZONE				
YEARS	SYSID	PRXTYPE	PRXPNAME	DEXUSER	DEXACCNT
	DEXGRP	DEXMBR	DEXTERM	YEAR	ZONE
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XDWMY.E	DEXACCNT	- Account Name
XDWMY.E	DEXGRP	- User Identification Code Group
XDWMY.E	DEXMBR	- User Identification Code Member
XDWMY.E	DEXTERM	- Terminal Name
XDWMY.E	DEXUSER	- Username
XDW...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year
XDWMY.E	PRXPNAME	- Process Name

```
 XDWMY.E  PRXTYPE  - Process Type
 XDWMY.E  SYSID    - System Identifier
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone

Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 XDWMY.   DEXBOOTS - Monitor System Boot Time Stamp
 XDWMY.   DEXDUR   - Requested Recording Interval Time
 X.....   DEXINTNO - Reserved Field Incremented by Monitor
 X.....   DEXMONVR - VMS Monitor Version
 X.....E  DEXPID   - Process ID
 XDWMY.   DEXTS    - Monitor Time Stamp
 XDWMY.E  DURATION - Recording Interval Time
 XDWMY.E  ENDTS    - End Time Stamp
 XDWMY.E  INTERVLS - Number of Recording Intervals
 X.....E  ORGSYSID - Originating System Identification
 XDWMY.E  STARTTS  - Start Time Stamp

Retained Data Elements

 X.....   PRXEVNTF - Event Flag Wait Mask
 X.....   PRXGLBPG - Global Page Count
 X.....   PRXGRP   - User Identification Code Group
 X.....   PRXIPID  - Internal Process ID
 X.....   PRXMBR   - User Identification Code Member
 X.....   PRXPRCPG - Process Page Count
 X.....   PRXPRTY  - Priority
 X.....   PRXSTATE - State
 X.....   PRXSTATF - Status Flags
 X.....   PRXWKSET - Process Working Set

Accumulated Data Elements

 XDWMY.E  PRXBUFIO - Buffered I/Os
 XDWMY.E  PRXCPUNI - Number of Instructions Executed
 XDWMY.E  PRXCPUTM - CPU Time
 XDWMY.E  PRXDIRIO - Direct I/Os
 XDWMY.E  PRXPGFLT - Page Faults

Maximum Data Elements

 XDWMY.E  PRXMXGPG - Maximum Global Page Count
 XDWMY.E  PRXMXPPG - Maximum Process Page Count
```

```
+-----------------+
| P R X B U F I O |  Buffered I/Os
+-----------------+
PRXBUFIO contains the count of I/O requests to buffered
devices, such as terminals and printers.

TYPE:  Accumulated - Numeric, measurement unit is integer.

SEE ALSO:  PROBUFIO - Buffered I/Os
           SPRBIOS  - Buffered I/Os
           PRCBUFIO - Buffered I/Os
           IMGBUFIO - Buffered I/Os
           INTBUFIO - Buffered I/Os

ELEMENT DERIVATION: PRXBUFIO is set to value of the PROBUFIO
element of the VMS MONITOR PROCESS File.

RANGE OF VALUE:  PRXBUFIO takes any numeric value in the
range from 0 to 2,147,483,647.

SUMMARIZATION PROCESS:  PRXBUFIO contains the sum of all
values of PRXBUFIO processed in the level of summarization.

USAGE NOTES:  PRXBUFIO reflects the number of buffered I/O
operations, not the size of those operations.


+-----------------+
| P R X C P U N I |  Number of Instructions Executed
+-----------------+
PRXCPUNI contains an estimate of the total number of CPU
instructions executed for a process, calculated by
multiplying the measured CPU time by a user-supplied VUPS
(where VAX Units of Power for VAX 11/780=1) rate.

TYPE:  Accumulated data element - Numeric, measurement unit
is integer.

SEE ALSO: PROCPUNI - Number of Instructions Executed

ELEMENT DERIVATION: PROCPUTM*MIPS

RANGE OF VALUE:  PRXCPUNI ranges from a minimum of 0 to a
maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS:  PRXCPUNI contains the sum of all
occurrences of PRXCPUNI that were included in the level of
summarization.

USAGE NOTES:  PRXCPUNI is an approximation based on system
engineering concepts that were valid for second-generation
```

equipment with hard-wired instructions and single-threaded execution. The accuracy of this element is a function of the instruction mix used in the program, the contending load (if any) using the system concurrently with the step, the processor architecture, the system microcode level, and the MIPS rate supplied by the user during the CA MICS generation process. Altering any of these factors will change the reported value of PRXCPUNI for replicate runs, even though the exact code executes each time a step is initiated.

When used with caution, PRXCPUNI can be helpful in comparing the relative efficiency of different software realizations of the same function (e.g., the relative path lengths of two database management systems for a backup/reload cycle performed on equivalent collections of data).

P R X C P U T M | CPU Time

PRXCPUTM contains the amount of CPU time used by the process for DURATION (measurement interval).

TYPE: Accumulated - Numeric, measurement unit is SAS time value.

SEE ALSO: PROCPUTM - CPU Time
PRCCPUTM - Total Process CPU Time
IMGCPUTM - Total Image CPU Time
INTCPUTM - CPU Time

ELEMENT DERIVATION: PRXCPUTM is set to the value of PROCPUTM of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXCPUTM ranges from 0 to the size of the DURATION (measurement interval) value.

SUMMARIZATION PROCESS: PRXCPUTM contains the sum of all values of PRXCPUTM processed in the level of summarization.

USAGE NOTES: When the process is swapped out, this value is set to zero. PRXCPUTM includes time spent only in user and supervisor modes. If the system spends CPU time in kernel or executive mode on behalf of a process, but not under the process' direct control, the CPU time is not included in this field.

P R X D I R I O | Direct I/Os

PRXDIRIO contains the count of I/O requests to nonbuffered devices, such as disks or tape drives.

TYPE: Accumulated - Numeric, measurement unit is integer.

SEE ALSO: PRODIRIO - Direct I/Os
SPRDIOS - Direct I/Os
PRCDIRIO - Direct I/Os
IMGDIRIO - Direct I/Os
INTDIRIO - Direct I/Os

ELEMENT DERIVATION: PRXDIRIO is set to the value of PRODIRIO of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXDIRIO takes any numeric value in the range from 0 to 2,147,483,647.

SUMMARIZATION PROCESS: PRXDIRIO contains the sum of all values of PRXDIRIO processed in the level of summarization.

USAGE NOTES: None

P R X E V N T F | Event Flag Wait Mask

PRXEVNTF contains the event flag wait bit mask of the process.

TYPE: Retained - Bit mask, 32 bits.

SEE ALSO: PROEVNTF - Event Flag Wait Mask

ELEMENT DERIVATION: PRXEVNTF is set to the value of PROEVNTF of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXEVNTF takes any hexadecimal value from 0 to FFFFFFFF.

SUMMARIZATION PROCESS: PRXEVNTF contains the last value of PRXEVNTF processed in the level of summarization.

USAGE NOTES: None

| P R X G L B P G | Global Page Count

PRXGLBPG contains the count of global pages mapped to this process.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROGLBPG - Global Page Count
PROMXGPG - Maximum Global Page Count
PRXMXGPG - Maximum Global Page Count

ELEMENT DERIVATION: PRXGLBPG is set to the value of PROGLBPG of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXGLBPG takes any numeric value from 0 to the number of global pages defined in VMS SYSGEN parameters.

SUMMARIZATION PROCESS: PRXGLBPG contains the last value of PRXGLBPG processed in the level of summarization.

USAGE NOTES: PRXGLBPG indicates how much shared memory the process is using. A high value in PRXGLBPG usually indicates a large amount of interprocess communication. Use of shared memory requires synchronization among processes to serialize access to the memory.

| P R X G R P | User Identification Code Group

PRXGRP contains the high-order word of the user identification code for the owner of the process. These values are assigned by the VAX system manager.

TYPE: Retained - Numeric, measurement unit is octal integer.

SEE ALSO: PROGRP - User Identification Code Group
PROMBR - User Identification Code Member
PRXMBR - User Identification Code Member
DEXGRP - User Identification Code Group

ELEMENT DERIVATION: PRXGRP is set to the value of PROGRP of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXGRP takes an octal integer value in the range 1 to 77777.

SUMMARIZATION PROCESS: PRXGRP contains the last value of PRXGRP processed in the level of summarization.

USAGE NOTES: PRXGRP can be used to summarize DETAIL records by group number. This summarization can show whether certain groups of users use resources at different levels from other groups. By itself, this value represents a class of VMS users; when combined with PRXMBR, it is a unique identifier for an authorized VMS account and makes up the user identification code. If the value is between 1 and 37776, PRXGRP indicates that the process is related to a user account. If the value is between 37777 and 77777, PRXGRP indicates that a resource identifier (assigned by the VAX system manager) owns the process.

| P R X I P I D | Internal Process ID

PRXIPID contains the internal process ID number assigned by VMS during process creation.

TYPE: Retained - Numeric, hexadecimal number unique on a system at any given time.

SEE ALSO: PROPID - Process ID
DEXPID - Process ID
PROEID - Extended Process ID

ELEMENT DERIVATION: PRXIPID is set to the value of PROPID of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXIPID takes any hexadecimal numeric value in the range from 00000001 to FFFFFFFF.

SUMMARIZATION PROCESS: PRXIPID contains the last value of PRXIPID processed in the level of summarization.

USAGE NOTES: The DEXPID number in the DETAIL time-span of the VMS ACCOUNTING files can be used to coordinate information with the VMS MONITOR data element PROEID. The PROPID number recorded in PRXIPID is local to VMS MONITOR PROCESSES record and is not used anywhere else.

| P R X M B R | User Identification Code Member

PRXMBR contains the low-order word of the user identification code for the owner of the process. These values are assigned by the VAX system manager.

TYPE: Retained - Numeric, measurement unit is octal integer.

SEE ALSO: PRXGRP - User Identification Code Group
PROMBR - User Identification Code Member
DEXMBR - User Identification Code Member

ELEMENT DERIVATION: PRXMBR is set to the value of the PROMBR element of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXMBR takes an octal integer value in the range 1 to 177777.

SUMMARIZATION PROCESS: PRXMBR contains the last value of PRXMBR processed in the level of summarization.

USAGE NOTES: PRXMBR should be used only in conjunction with PRXGRP to identify each unique VMS user. When combined with PRXGRP, this is a unique identifier for authorized user accounts and makes up the user identification code. If the value is 177777, PRXMBR indicates that the owner of the process is a resource identifier rather than a user account.

P R X M X G P G | Maximum Global Page Count

PRXMXGPG contains the maximum number of global pages used by the process.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRXGLBPG - Global Page Count
PROGLBPG - Global Page Count

ELEMENT DERIVATION: PRXMXGPG is set to the maximum value of PRXGLBPG reached.

RANGE OF VALUE: PRXMXGPG takes any integer value from 0 to the number of global pages defined in VMS SYSGEN parameters.

SUMMARIZATION PROCESS: PRXMXGPG contains the maximum value of PRXMXGPG processed for the level of summarization.

USAGE NOTES: PRXMXGPG indicates the maximum shared memory the process used. A high value in PRXMXGPG usually indicates a large amount of interprocess communication. Use of shared memory requires synchronization among processes to serialize access to the memory.

P R X M X P P G | Maximum Process Page Count

PRXMXPPG contains the maximum number of local pages of memory used by the process.

TYPE: Maximum - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRXPRCPG - Process Page Count
PROPRCPG - Process Page Count

ELEMENT DERIVATION: PRXMXPPG is set to the maximum value reached by PRXPRCPG.

RANGE OF VALUE: PRXMXPPG takes any integer value from 0 to the page file quota set in the user account.

SUMMARIZATION PROCESS: PRXMXPPG contains the maximum value of PRXMXPPG processed for the level of summarization.

USAGE NOTES: PRXMXPPG is 0 for the SWAPPER process. If PRXMXPPG is larger than the user's working set size, or if it is larger than the working set quota and the system has run out of free pages, the process is subject to page faulting.

P R X P G F L T | Page Faults

PRXPGFLT contains the count of page faults for the duration of the interval.

TYPE: Accumulated - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PROPGFLT - Page Faults
PRCPAGFT - Page Faults
IMGPAGFT - Page Faults
SPRPGEFL - PGE Page Faults

ELEMENT DERIVATION: PRXPGFLT is set to the value of PROPGFLT of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXPGFLT takes any integer value from 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: PRXPGFLT contains the sum of all values of PRXPGFLT processed in the level of summarization.

USAGE NOTES: A high value for PRXPGFLT means that the

process is not finding the pages of process data or program memory resident in the physical memory of the machine when it needs them. PRXPGFLT is high for two primary reasons:

- o The process is continually activating new images.
- o The images are larger than the allowed working set.

Large amounts of page faulting will degrade system performance. If the size of physical memory permits, some page faulting can be reduced by increasing the working set size of the processes that tend to have high values in PRXPGFLT.

P R X P N A M E Process Name

PRXPNAME contains the name of the process.

TYPE: Retained - Character, length 15.

SEE ALSO: PROPNAME - Process Name

ELEMENT DERIVATION: PRXPNAME is set to the value of PROPNAME of the VMS MONITOR PROCESS file. Source data contains counted character string, 1 byte for count, 15 bytes of characters.

RANGE OF VALUE: PRXPNAME can contain any combination of 1 to 15 characters. The count byte from source data is not stored.

SUMMARIZATION PROCESS: PRXPNAME is not summarized. PRXPNAME is used to determine a unique key for records within a file based on the time-span, so other elements in the file can be summarized.

USAGE NOTES: None

P R X P R C P G Process Page Count

PRXPRCPG contains the number of local pages of memory in use by the process.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRXMXPPG - Maximum Process Page Count

PROPRCPG - Process Page Count

ELEMENT DERIVATION: PRXPRCPG is set to the value of PROPRCPG of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXPRCPG takes any integer value from 0 to the page file quota set in the user account.

SUMMARIZATION PROCESS: PRXPRCPG contains the last value of PRXPRCPG processed in the level of summarization.

USAGE NOTES: PRXPRCPG is 0 for the SWAPPER process. If PRXPRCPG is larger than the user's working set size, or if it is larger than the working set quota and the system has run out of free pages, the process will begin page faulting. Processes with a large PRXPRCPG are the first targets for swapping.

If the sum of PRXPRCPG for all processes at a given time exceeds the size of the page file, VMS will stop the process that attempted to exceed the page file size. If the page file is fragmented, VMS may halt the system.

P R X P R T Y Priority

PRXPRTY contains the priority level code for the process.

TYPE: Retained - Numeric, measurement unit is integer, 1 byte long.

SEE ALSO: PROPRTY - Priority
DEXPRTY - Priority

ELEMENT DERIVATION: PRXPRTY is set to the value of PROPRTY of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXPRTY can be any integer value from 0 to 30.

SUMMARIZATION PROCESS: PRXPRTY contains the last value of PRXPRTY processed in the level of summarization.

USAGE NOTES: PRXPRTY is the actual priority level subtracted from 31. If PRXPRTY is 15 or less, the process is running in real-time mode and cannot be preempted by the scheduler. On most systems, SWAPPER is the only process running at that high a priority; its PRXPRTY is 15. Larger values of PRXPRTY mean lower priority for a process.

For processes with PRXPRTY 16 or greater, the scheduler decreases PRXPRTY each time it cannot schedule the process to run. When a process runs the scheduler, it sets PRXPRTY to the base level for that process. The scheduler will never cause PRXPRTY to become less than 16.

P R X S T A T E | State

PRXSTATE contains a code for the processing state of the process.

TYPE: Retained - Numeric, measurement unit is integer, 2 bytes long.

SEE ALSO:
- PROSTATE - State
- SPRCOLPG - Collided Page Wait
- SPRMWAIT - Miscellaneous Resource Wait
- SPRCEF - Common Event Flag Wait
- SPRPFW - Page Fault Wait
- SPRLEF - Local Event Flag Wait, Inswapped
- SPRLEFO - Local Event Flag Wait, Outswapped
- SPRHIB - Hibernate Wait, Inswapped
- SPRHIBO - Hibernate Wait, Outswapped
- SPRSUSP - Suspended Wait, Inswapped
- SPRSUSPO - Suspended Wait, Outswapped
- SPRFPGWT - Free Page Wait
- SPRCOM - Compute State Wait, Inswapped
- SPRCOMO - Compute State Wait, Outswapped
- SPRCUR - Current Processes

ELEMENT DERIVATION: PRXSTATE is set to the value of PROSTATE of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXSTATE takes any integer value from 1 to 14.

SUMMARIZATION PROCESS: PRXSTATE contains the last value of PRXSTATE processed in the level of summarization.

USAGE NOTES: PRXSTATE is currently defined for 14 values, although DEC may define more values in the future. Each value and its meaning follow:

1 - COLPG collided page wait
2 - MWAIT miscellaneous resource wait
3 - CEF common event flag wait
4 - PFW page fault wait
5 - LEF local event flag wait, inswapped
6 - LEFO local event flag wait, outswapped
7 - HIB hibernate wait, inswapped
8 - HIBO hibernate wait, outswapped
9 - SUSP suspended wait, inswapped
10 - SUSPO suspended wait, outswapped
11 - FPGWT free page wait
12 - COM compute state wait, inswapped
13 - COMO compute state wait, outswapped
14 - CUR current processes

If a process is in either the HIB or HIBO state, it is waiting for some time to pass, and it is not affecting system performance nor is system activity affecting it. If a process is in either the LEF or LEFO state, it is probably waiting for user input and is also not affecting the system negatively. If a process is in either the SUSP or SUSPO state, another process has suspended its operation. A process in the COLPG, MWAIT, CEF, PFW, or FPGWT state is waiting for some other process to free some resource(s) before continuing.

Processes in LEFO, HIBO, SUSPO, and COMO states indicate that the system probably does not have enough memory for the work that it is doing. Processes in the COM and COMO states indicate that the CPU is not fast enough to keep up with the workload. Many processes in the COLPG, PFW, and FPGWT states indicate that working sets may not be large enough.

P R X S T A T F | Status Flags

PRXSTATF contains the status flag indicators for the process.

TYPE: Retained - Numeric, measurement unit is hexadecimal integer, 4 bytes long.

SEE ALSO: PROSTATF - Status Flags

ELEMENT DERIVATION: PRXSTATF is set to the value of PROSTATF of the VMS MONITOR PROCESS file.

RANGE OF VALUE: PRXSTATF takes any hexadecimal integer value in the range from 00000000 to FFFFFFFF.

SUMMARIZATION PROCESS: PRXSTATF contains the last value of PRXSTATF processed in the level of summarization.

USAGE NOTES: Bit 0 (PCB$V_RES) of PRXSTATF is set to 1 when the process is in memory and 0 when it is swapped out. The recording intervals possible through VMS MONITOR, however, are not small enough to make PRXSTATF very useful to VAX

system managers.

| P R X T Y P E | Process Type

The process type for remote process.

TYPE: Common data element - Character, length 8.

SEE ALSO: PRCTYPE - Process Type

ELEMENT DERIVATION: PRXTYPE is obtained from PRCTYPE of the VMS ACCOUNTING Process Termination file, for those processes that can be matched between the MONITOR and ACCOUNTING data sources.

RANGE OF VALUE: This element may contain any of the following values:

```
 o 'PRINTJOB'    /* 'PRINT JOB'             */
 o 'INTERACT'    /* 'INTERACTIVE PROCESS'   */
 o 'SUBPROC '    /* 'SUB PROCESS'           */
 o 'DETACHED'    /* 'DETACHED PROCESS'      */
 o 'BATCH   '    /* 'BATCH PROCESS'         */
 o 'NETWORK '    /* 'NETWORK PROCESS'       */
 o 'UNKNOWN '
```

SUMMARIZATION PROCESS: PRXTYPE contains the value of the last occurrence of PRXTYPE in the observations processed for the summarization level.

USAGE NOTES: None

| P R X W K S E T | Process Working Set

PRXWKSET contains the working set size of the process.

TYPE: Retained - Numeric, measurement unit is integer, 4 bytes long.

SEE ALSO: PRCWKSPK - Working Set Peak
IMGWKSPK - Working Set Peak

ELEMENT DERIVATION: PRXWKSET is set to the sum of the values: PROGLBPG+PROPRCPG, from the VMS MONITOR Process file.

RANGE OF VALUE: PRXWKSET takes any integer value from 1 to the smallest value specified by the SYSGEN parameter WSMAX or the user authorization file parameter WSEXTENT.

SUMMARIZATION PROCESS: PRXWKSET contains the last value of PRXWKSET processed in the level of summarization.

USAGE NOTES: If PRXWKSET is smaller than WSQUOTA for the user, all page faulting was a result of image activation. If PRXWKSET is nearly as large as WSEXTENT, the process was borrowing pages from free memory. When PRXWKSET is growing to nearly the size of WSEXTENT, it indicates that there is sufficient memory free and the system is not swapping. If paging is high and this condition occurs, adjusting WSQUOTA and WSEXTENT higher may improve overall performance.

B.4.6 - The VMS System Status (DESSYU) File

FILE ABSTRACT: VMS System Status File	
CONTENT	Records a number of system readings for each execution of the CA MICS DEXSUS collection routine.
USE	To identify hardware characteristics and system settings for clarity of reporting and to augment memory use measurements for performance tuning and capacity planning.
SOURCE	VMS MONITOR System Information record, type 129, VMS SHOW SYSTEM and SHOW MEMORY output, and several VMS lexical functions, including F$GETSYI.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	YEAR	MONTH	DAY	ENDTS
DAYS	SYSID	YEAR	MONTH	DAY	
WEEKS	SYSID	YEAR	WEEK		
MONTHS	SYSID	YEAR	MONTH		
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Jul 18, 2019

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Jul 18, 2019

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
Sequence/Summary Data Elements		
XD....E	DAY	- Day of Month
XD.M..E	MONTH	- Month of Year
XDWM..E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWM..E	YEAR	- Year of Century
Common Data Elements		
XD....E	DAYNAME	- Name of Day of Week
X.....E	DEXNODE	- Nodename
XDWM..E	DEXSUSTB	- DEXSUS System Boot Time Stamp
XDWM..E	DEXSUSTS	- DEXSUS Collect Time Stamp
XDWM..E	ENDTS	- End Time Stamp
XDW...E	HOUR	- Hour of Day
X.....E	ORGSYSID	- Originating System Identification
XDWM..E	ZONE	- Time Zone

Retained Data Elements

```
 XDWM..E  SYUBALSM - Process Pages for Memory Allocation
 XDWM..E  SYUCPUS  - Number of CPUs
 XDWM..E  SYUHWNAM - Hardware Name
 XDWM..E  SYUIRPC  - IRP Lookaside List, Current Packets
 XDWM..E  SYUIRPI  - IRP Lookaside List, Initial Packets
 XDWM..E  SYUIRPM  - IRP Lookaside List, Maximum Packets
 XDWM..E  SYULRPC  - LRP Lookaside List, Current Packets
 XDWM..E  SYULRPI  - LRP Lookaside List, Initial Packets
 XDWM..E  SYULRPM  - LRP Lookaside List, Maximum Packets
 XDWM..E  SYUMEMAN - Main Memory Mb (physical)
 XDWM..E  SYUMENDC - Nonpaged Dynamic Memory, Current Size
 XDWM..E  SYUMENDI - Nonpaged Dynamic Memory, Initial Size
 XDWM..E  SYUMEPDC - Paged Dynamic Memory, Current Size
 XDWM..E  SYUMPWHI - MPW_HILIMIT System Parameter
 XDWM..E  SYUMXPRC - MAXPROCESSCNT System Parameter
 XDWM..E  SYUSRPC  - SRP Lookaside List, Current Packets
 XDWM..E  SYUSRPI  - SRP Lookaside List, Initial Packets
 XDWM..E  SYUSRPM  - SRP Lookaside List, Maximum Packets
 XDWM..E  SYUUPTIM - System Uptime
 XDWM..E  SYUVCPUS - Number of Vector-Present Processors
 XDWM..E  SYUVMSVR - VMS Version
```

Maximum Data Elements

```
 XDWM..E  SYUMXMEN - Nonpaged Dynamic Memory, Maximum Size
```

```
+-----------------+
| S Y U B A L S M |  Process Pages for Memory Allocation
+-----------------+
```

SYUBALSM contains the number of process pages to which memory can be allocated, measured at VMS Monitor startup.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SYUBALSM is obtained from VMS Monitor System Information record (type 129), from MNR_SYI$L_BALSETMEM.

RANGE OF VALUE: SYUBALSM takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUBALSM contains the last value of SYUBALSM processed in the level of summarization.

USAGE NOTES: SYUBALSM is obtained during daily input processing from the MONITOR data source. The measurement is recorded to the SYU file if a DEXSUS collect event with a time stamp greater than MONITOR start time is processed. The DEXSUS time stamp is recorded in ENDTS.

```
+-----------------+
| S Y U C P U S   |  Number of CPUs
+-----------------+
```

SYUCPUS contains count of available CPUs.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: SYUVCPUS - Number of Vector-Present Processors

ELEMENT DERIVATION: SYUCPUS is set to the value of the AVAILCPU_CNT field from the DEXSUS SYI4 separator record.

RANGE OF VALUE: SYUCPUS takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUCPUS contains the last value of SYUCPUS processed in the level of summarization.

USAGE NOTES: This element is set from the MONITOR type 129 record if not available otherwise.

```
+-----------------+
| S Y U H W N A M |  Hardware Name
+-----------------+
```

SYUHWNAM contains the hardware name of the VAX system.

TYPE: Retained - Character, length 48.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SYUHWNAM is set to the value of the HW_NAME field from the DEXSUS SYI1 separator record.

RANGE OF VALUE: SYUHWNAM can contain any combination of 1 to 48 alphanumeric characters.

SUMMARIZATION PROCESS: SYUHWNAM contains the last value of SYUHWNAM processed in the level of summarization.

USAGE NOTES: None

```
+-----------------+
| S Y U I R P C   |  IRP Lookaside List, Current Packets
+-----------------+
SYUIRPC contains the current total size for the I/O Request
Packet (IRP) Lookaside List.

This element will have missing values for systems running VMS
Version 6.0 or later.

TYPE:  Retained - Numeric, measurement unit is integer.

SEE ALSO:  SYUIRPI - IRP Lookaside List, Initial Packets
           SYUIRPM - IRP Lookaside List, Maximum Packets

ELEMENT DERIVATION:  SYUIRPC is set to the IRP packets value
for "Current Total Size" reported by SHOW MEM/FULL.  This
information is collected in the DEXSUS MEM1 measurement
records set.

RANGE OF VALUE:  SYUIRPC takes any numeric value in the range
from a minimum of 0 to a maximum that is limited only by
practical bounds.

SUMMARIZATION PROCESS:  SYUIRPC contains the last value of
SYUIRPC processed in the level of summarization.

USAGE NOTES:  None


+-----------------+
| S Y U I R P I   |  IRP Lookaside List, Initial Packets
+-----------------+
SYUIRPI contains the initial size (IRPCOUNT) for the I/O
Request Packet (IRP) Lookaside List.

This element will have missing values for systems running VMS
Version 6.0 or later.

TYPE:  Retained - Numeric, measurement unit is integer.

SEE ALSO:  SYUIRPC - IRP Lookaside List, Current Packets
           SYUIRPM - IRP Lookaside List, Maximum Packets

ELEMENT DERIVATION:  SYUIRPI is set to the IRP packets value
for "Initial Size" reported by SHOW MEM/FULL.  This
information is collected in the DEXSUS MEM1 measurement
records set.

RANGE OF VALUE:  SYUIRPI takes any numeric value in the range
from a minimum of 0 to a maximum that is limited only by
practical bounds.

SUMMARIZATION PROCESS:  SYUIRPI contains the last value of
SYUIRPI processed in the level of summarization.

USAGE NOTES:  None


+-----------------+
| S Y U I R P M   |  IRP Lookaside List, Maximum Packets
+-----------------+
SYUIRPM contains the maximum size (IRPCOUNTV) for the I/O
Request Packet (IRP) Lookaside List.

This element will have missing values for systems running VMS
Version 6.0 or later.

TYPE:  Retained - Numeric, measurement unit is integer.

SEE ALSO:  SYUIRPC - IRP Lookaside List, Current Packets
           SYUIRPI - IRP Lookaside List, Initial Packets

ELEMENT DERIVATION:  SYUIRPM is set to the IRP packets value
for "Maximum Size" reported by SHOW MEM/FULL.  This
information is collected in the DEXSUS MEM1 measurement
records set.

RANGE OF VALUE:  SYUIRPM takes any numeric value in the range
from a minimum of 0 to a maximum that is limited only by
practical bounds.

SUMMARIZATION PROCESS:  SYUIRPM contains the last value of
SYUIRPM processed in the level of summarization.

USAGE NOTES:  None


+-----------------+
| S Y U L R P C   |  LRP Lookaside List, Current Packets
+-----------------+
SYULRPC contains the current total size for the Large Packet
(LRP) Lookaside List.

This element will have missing values for systems running VMS
Version 6.0 or later.

TYPE:  Retained - Numeric, measurement unit is integer.

SEE ALSO:  SYULRPI - LRP Lookaside List, Initial Packets
           SYULRPM - LRP Lookaside List, Maximum Packets

ELEMENT DERIVATION:  SYULRPC is set to the LRP packets value
for "Current Total Size" reported by SHOW MEM/FULL.  This
information is collected in the DEXSUS MEM1 measurement
```

records set.

RANGE OF VALUE: SYULRPC takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYULRPC contains the last value of SYULRPC processed in the level of summarization.

USAGE NOTES: None

S Y U L R P I — LRP Lookaside List, Initial Packets

SYULRPI contains the initial size (LRPCOUNT) for the Large Packet (LRP) Lookaside List.

This element will have missing values for systems running VMS Version 6.0 or later.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: SYULRPC - LRP Lookaside List, Current Packets
SYULRPM - LRP Lookaside List, Maximum Packets

ELEMENT DERIVATION: SYULRPI is set to the LRP packets value for "Initial Size" reported by SHOW MEM/FULL. This information is collected in the DEXSUS MEM1 measurement records set.

RANGE OF VALUE: SYULRPI takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYULRPI contains the last value of SYULRPI processed in the level of summarization.

USAGE NOTES: None

S Y U L R P M — LRP Lookaside List, Maximum Packets

SYULRPM contains the maximum size (LRPCOUNTV) for the Large Packet (LRP) Lookaside List.

This element will have missing values for systems running VMS Version 6.0 or later.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: SYULRPC - LRP Lookaside List, Current Packets
SYULRPI - LRP Lookaside List, Initial Packets

ELEMENT DERIVATION: SYULRPM is set to the LRP packets value for "Maximum Size" reported by SHOW MEM/FULL. This information is collected in the DEXSUS MEM1 measurement records set.

RANGE OF VALUE: SYULRPM takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYULRPM contains the last value of SYULRPM processed in the level of summarization.

USAGE NOTES: None

S Y U M E M A N — Main Memory Mb (physical)

SYUMEMAN contains the amount of physical memory on the VAX system.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SYUMEMAN is set to the main memory value for "Physical Memory" reported by SHOW MEM/PHYS. This information is collected in the DEXSUS MEM2 measurement records set.

RANGE OF VALUE: SYUMEMAN takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUMEMAN contains the last value of SYUMEMAN processed in the level of summarization.

USAGE NOTES: None

S Y U M E N D C — Nonpaged Dynamic Memory, Current Size

SYUMENDC contains the current size in bytes for Nonpaged Dynamic Memory.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: SYUMENDI - Nonpaged Dynamic Memory, Initial Size

SYUMXMEN - Nonpaged Dynamic Memory, Maximum Size

ELEMENT DERIVATION: SYUMENDC is set to the Nonpaged Dynamic Memory value for "Current Size" reported by SHOW MEM/FULL. This information is collected in the DEXSUS MEM1 measurement records set.

RANGE OF VALUE: SYUMENDC takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUMENDC contains the last value of SYUMENDC processed in the level of summarization.

USAGE NOTES: None

S Y U M E N D I — Nonpaged Dynamic Memory, Initial Size

SYUMENDI contains the initial size in bytes for Nonpaged Dynamic Memory (NPAGEDYN).

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: SYUMENDC - Nonpaged Dynamic Memory, Current Size
SYUMXMEN - Nonpaged Dynamic Memory, Maximum Size

ELEMENT DERIVATION: SYUMENDI is set to the Nonpaged Dynamic Memory value for "Initial Size" reported by SHOW MEM/FULL. This information is collected in the DEXSUS MEM1 measurement records set.

RANGE OF VALUE: SYUMENDI takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUMENDI contains the last value of SYUMENDI processed in the level of summarization.

USAGE NOTES: None

S Y U M E P D C — Paged Dynamic Memory, Current Size

SYUMEPDC contains the current size in bytes for Paged Dynamic Memory (PAGEDYN).

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: SYUMENDI - Nonpaged Dynamic Memory, Initial Size
SYUMXMEN - Nonpaged Dynamic Memory, Maximum Size
SYUMENDC - Nonpaged Dynamic Memory, Current Size

ELEMENT DERIVATION: SYUMEPDC is set to the Paged Dynamic Memory value for "Current Size" reported by SHOW MEM/FULL. This information is collected in the DEXSUS MEM1 measurement records set.

RANGE OF VALUE: SYUMEPDC takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUMEPDC contains the last value of SYUMEPDC processed in the level of summarization.

USAGE NOTES: None

S Y U M P W H I — MPW_HILIMIT System Parameter

SYUMPWHI contains the MPW_HILIMIT System Parameter value.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SYUMPWHI is obtained from VMS Monitor System Information record (type 129), from MNR_SYI$L_MPWHILIM.

RANGE OF VALUE: SYUMPWHI takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUMPWHI contains the last value of SYUMPWHI processed in the level of summarization.

USAGE NOTES: SYUMPWHI is obtained during daily input processing from the MONITOR data source. The measurement is recorded to the SYU file if a DEXSUS collect event with a time stamp greater than MONITOR start time is processed. The DEXSUS time stamp is recorded in ENDTS.

S Y U M X M E N — Nonpaged Dynamic Memory, Maximum Size

SYUMXMEN contains the maximum size in bytes for Nonpaged Dynamic Memory (NPAGEVIR).

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: SYUMENDC - Nonpaged Dynamic Memory, Current Size
SYUMENDI - Nonpaged Dynamic Memory, Initial Size

ELEMENT DERIVATION: SYUMXMEN is set to the Nonpaged Dynamic Memory value for "Maximum Size" reported by SHOW MEM/FULL. This information is collected in the DEXSUS MEM1 measurement records set.

RANGE OF VALUE: SYUMXMEN takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUMXMEN contains the maximum value of SYUMXMEN processed for the level of summarization.

USAGE NOTES: None

S Y U M X P R C — MAXPROCESSCNT System Parameter

SYUMXPRC contains the MAXPROCESSCNT System Parameter value.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SYUMXPRC is obtained from VMS Monitor System Information record (type 129), from MNR_SYI$W_MAXPRCCNT.

RANGE OF VALUE: SYUMXPRC takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUMXPRC contains the last value of SYUMXPRC processed in the level of summarization.

USAGE NOTES: SYUMXPRC is obtained during daily input processing from the MONITOR data source. The measurement is recorded to the SYU file if a DEXSUS collect event with a time stamp greater than MONITOR start time is processed. The DEXSUS time stamp is recorded in ENDTS.

S Y U S R P C — SRP Lookaside List, Current Packets

SYUSRPC contains the current total size for the Small Packet (SRP) Lookaside List.

This element will have missing values for systems running VMS Version 6.0 or later.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: SYUSRPI - SRP Lookaside List, Initial Packets
SYUSRPM - SRP Lookaside List, Maximum Packets

ELEMENT DERIVATION: SYUSRPC is set to the SRP packets value for "Current Total Size" reported by SHOW MEM/FULL. This information is collected in the DEXSUS MEM1 measurement records set.

RANGE OF VALUE: SYUSRPC takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUSRPC contains the last value of SYUSRPC processed in the level of summarization.

USAGE NOTES: None

S Y U S R P I — SRP Lookaside List, Initial Packets

SYUSRPI contains the initial size (SRPCOUNT) for the Small Packet (SRP) Lookaside List.

This element will have missing values for systems running VMS Version 6.0 or later.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: SYUSRPC - SRP Lookaside List, Current Packets
SYUSRPM - SRP Lookaside List, Maximum Packets

ELEMENT DERIVATION: SYUSRPI is set to the SRP packets value for "Initial Size" reported by SHOW MEM/FULL. This information is collected in the DEXSUS MEM1 measurement records set.

RANGE OF VALUE: SYUSRPI takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUSRPI contains the last value of SYUSRPI processed in the level of summarization.

USAGE NOTES: None

| S Y U S R P M | SRP Lookaside List, Maximum Packets

SYUSRPM contains the maximum size (SRPCOUNTV) for the Small Packet (SRP) Lookaside List.

This element will have missing values for systems running VMS Version 6.0 or later.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: SYUSRPC - SRP Lookaside List, Current Packets
SYUSRPI - SRP Lookaside List, Initial Packets

ELEMENT DERIVATION: SYUSRPM is set to the SRP packets value for "Maximum Size" reported by SHOW MEM/FULL. This information is collected in the DEXSUS MEM1 measurement records set.

RANGE OF VALUE: SYUSRPM takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUSRPM contains the last value of SYUSRPM processed in the level of summarization.

USAGE NOTES: None

| S Y U U P T I M | System Uptime

SYUUPTIM contains the amount of time since last system boot.

TYPE: Retained - Numeric, measurement unit is SAS time value.

SEE ALSO: DEXBOOTS - Monitor System Boot Time Stamp
DEXSUSTB - DEXSUS System Boot Time Stamp

ELEMENT DERIVATION: SYUUPTIM is set to the Uptime value reported by SHOW SYS. This information is collected in the DEXSUS SYS1 measurement records set.

RANGE OF VALUE: SYUUPTIM ranges from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUUPTIM contains the last value of SYUUPTIM processed in the level of summarization.

USAGE NOTES: None

| S Y U V C P U S | Number of Vector-Present Processors

SYUVCPUS contains number of Vector-Present processors on the system.

TYPE: Retained - Numeric, measurement unit is integer.

SEE ALSO: SYUCPUS - Number of CPUs

ELEMENT DERIVATION: SYUVCPUS is obtained from VMS Monitor System Information record (type 129), from MNR_SYI$B_VPCPUS.

RANGE OF VALUE: SYUVCPUS takes any numeric value in the range from a minimum of 0 to a maximum that is limited only by practical bounds.

SUMMARIZATION PROCESS: SYUVCPUS contains the last value of SYUVCPUS processed in the level of summarization.

USAGE NOTES: SYUVCPUS is obtained during daily input processing from the MONITOR data source. The measurement is recorded to the SYU file if a DEXSUS collect event with a time stamp greater than MONITOR start time is processed. The DEXSUS time stamp is recorded in ENDTS.

| S Y U V M S V R | VMS Version

SYUVMSVR contains VMS Version under which the DEXSUS collect was run.

TYPE: Retained - Character, length 8.

SEE ALSO: Not applicable

ELEMENT DERIVATION: SYUVMSVR is set to the value of the VERSION field from the DEXSUS SYI2 separator record.

RANGE OF VALUE: SYUVMSVR can contain any combination of 1 to 8 alphanumeric characters.

SUMMARIZATION PROCESS: SYUVMSVR contains the last value of SYUVMSVR processed in the level of summarization.

USAGE NOTES: None