

CA MICS[®] Resource Management

Analyzer Option for VAX/VMS Guide

Release 12.8.00



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "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 YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: OVERVIEW	9
1.1 Primary Areas of Application.....	12
1.2 Reporting and Inquiry Facilities.....	13
1.3 Files Overview	14
1.4 Product Prerequisites	18
Chapter 2: USAGE GUIDELINES	19
2.1 Data Analysis	20
2.2 CA MICS Product Interfaces	22
2.2.1 Accounting and Chargeback Interface	22
2.2.2 Capacity Planner Interface	23
Chapter 3: REPORTS	25
3.1 MICS Inquiries for VAX/VMS	26
3.1.1 VAX/VMS Workload Analysis Inquiries	29
3.1.2 VAX/VMS Performance Analysis Inquiries	39
3.1.3 VAX/VMS Disk Storage Analysis Inquiries	57
Chapter 4: EXCEPTIONS	75
Chapter 5: FILES	77
5.1 Data Element Naming Conventions	79
5.2 VAX/VMS Accounting Information Area Files	80
5.2.1 VMS ACCOUNTING Process Termination File (DEAPRC)	80
5.2.2 VMS ACCOUNTING Image Termination File (DEAIMG)	86
5.2.3 VMS ACCOUNTING LOGIN Failed File (DEA_LF)	92
5.2.4 VMS ACCOUNTING Print Queued File (DEAPRQ)	97
5.2.5 VMS ACCOUNTING Initialization File (DEAINT)	102
5.3 VAX/VMS Monitor Information Area Files	107
5.3.1 VMS MONITOR Process File (DEMPRO)	107
5.3.2 VMS MONITOR Modes File (DEMMOD)	114
5.3.3 VMS MONITOR Disk File (DEMDSK)	120
5.3.4 VMS MONITOR SCS File (DEMSCS)	126
5.3.5 VMS MONITOR RMS File (DEMRRMS).....	131
5.3.6 VMS System Profile File (DEMSPR)	138

5.4 VAX/VMS System Usage Information Area Files	159
5.4.1 VMS Disk Device File (DESDKD)	160
5.4.2 VMS Disk Quota File (DESDKQ)	164
5.4.3 VMS Disk Usage File (DESDKU).....	169
5.4.4 VMS Process Activity File (DESPRX).....	174
5.4.5 VMS System Status File (DESSYU)	181

Chapter 6: DATA SOURCES 187

6.1 Data Collection and Transport to CA MICS.....	187
6.1.1 Data Collector Operations.....	189
6.1.2 Data Volume Considerations.....	197
6.1.3 Transporting Data from VMS to MVS.....	200
6.1.4 Errors Identified in Measurement Data	201
6.2 VMS ACCOUNTING Record Descriptions	202
6.2.1 PROCESS TERMINATION Record	202
6.2.2 IMAGE TERMINATION Record	203
6.2.3 SYSTEM INITIALIZATION Record	204
6.2.4 LOGIN FAILED Record	205
6.2.5 PRINT QUEUED Record.....	206
6.3 VMS MONITOR Record Descriptions	207
6.3.1 FILE HEADER Record.....	207
6.3.2 SYSTEM INFORMATION Record.....	208
6.3.3 CLUSTER Record	208
6.3.4 DECNET Record	208
6.3.5 DISK Record.....	209
6.3.6 DLOCK Record	209
6.3.7 FCP Record	209
6.3.8 FILE SYSTEM CACHE Record	210
6.3.9 IO Record.....	210
6.3.10 LOCK Record	210
6.3.11 MODES Record	211
6.3.12 MSCP SERVER Record.....	211
6.3.13 PAGE Record	211
6.3.14 POOL Record	212
6.3.15 PROCESSES Record	212
6.3.16 RMS Record.....	212
6.3.17 SCS Record	213
6.3.18 STATES Record	213
6.3.19 SYSTEM Record	213
6.3.20 TRANSACTION Record.....	214
6.3.21 VECTOR Record	214

6.4 CA MICS VMS System Usage (DEXSUS) Record Descriptions	214
6.4.1 DEXSUS FILE HEADER Record	215
6.4.2 DEXSUS SYSTEM INFORMATION Records	216
6.4.3 DEXSUS SYI1 Record	216
6.4.4 DEXSUS SYI2 Record	216
6.4.5 DEXSUS SYI3 Record	217
6.4.6 DEXSUS SYI4 Record	217
6.4.7 DEXSUS SYSTEM MEMORY 1 Records	217
6.4.8 DEXSUS SYSTEM MEMORY 2 Records	218
6.4.9 DEXSUS DISK DEVICE Records	218
6.4.10 DEXSUS DISK QUOTA Records	218
6.4.11 DEXSUS DISK USAGE Records	219
6.4.12 DEXSUS FILE TRAILER Record	219

Chapter 7: DEFINING PARAMETERS **221**

7.1 Environmental Considerations	222
7.2 Complex Level Parameters	222
7.2.1 Analyzer Definition Statements (DEXGENIN)	222
7.2.2 VAX Account Code Structure (DEXACCT)	223
7.2.3 Code the VAX/VMS Account Code Exit (DEXACRT)	227
7.2.4 Code the VAX/VMS Disk Account Code Exit (DEXACRTD)	230
7.3 Unit Level Parameters	233
7.3.1 VAX/VMS Processing Options (DEXOPS)	234
7.3.2 Input DD Statements for VAX/VMS (INPUTDEX)	276
7.3.3 Input Needed to Generate VAX System Code (DEXPGEN)	279
7.3.4 Database Space Modeling (DBMODEL)	280
7.4 Other Related Parameters	284

Chapter 8: INSTALLATION **289**

8.1 Installing the DEXSUS.COM Collection Routine	289
--	-----

Chapter 9: PROCESSING **291**

9.1 Processing Considerations	291
9.2 Daily Update Processing Flow	292
9.2.1 Input Raw Data	292
9.2.2 Information Area Processing	295
9.2.3 Database Timespan Processing	296
9.2.4 File Aging	296
9.3 User Exits	296
9.4 Using COMMON Input DDnames and DEXINPUT exits	299

Chapter 10: MODIFICATION	305
10.1 Parameter Modification	305
10.1.1 Change the Number of Work Files for DAY074	306
10.1.2 Enable Internal Step Restart	309
10.1.3 Implement Incremental Update.....	311
Appendix A: MESSAGES	315
Messages.....	315
Appendix B: DATA DICTIONARY	343

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.

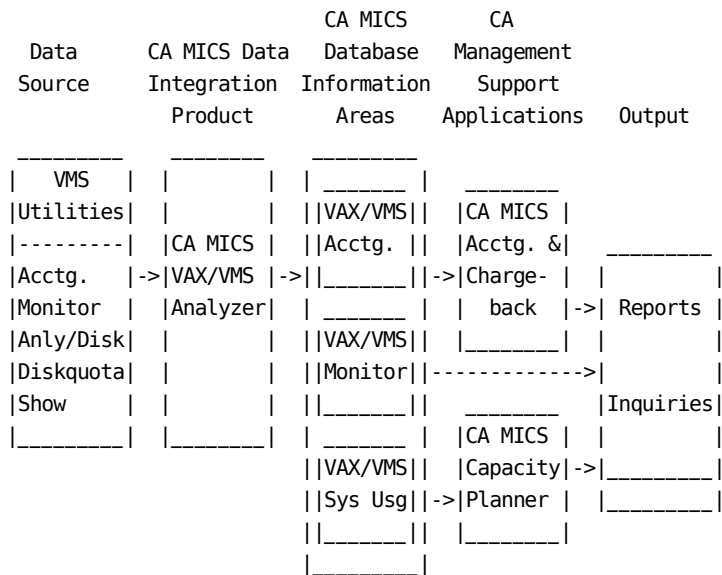


Figure 1-1. CA MICS VAX/VMS Analyzer Data Flow

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 section contains the following topics:

[1.1 Primary Areas of Application](#) (see page 12)

[1.2 Reporting and Inquiry Facilities](#) (see page 13)

[1.3 Files Overview](#) (see page 14)

[1.4 Product Prerequisites](#) (see page 18)

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:

- o 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.
- o 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.
- o 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.
- o 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.
- o 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.

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.

This section contains the following topics:

[2.1 Data Analysis](#) (see page 20)

[2.2 CA MICS Product Interfaces](#) (see page 22)

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.

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.

This section contains the following topics:

[3.1 MICF Inquiries for VAX/VMS](#) (see page 26)

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:

- 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.

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

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				SYSID = VMS2 Year = 08 Week = 25 Time Zone = 1				RUN DATE: 1AUG08	
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				SYSID = VMS2 Year = 08 Month = 6				RUN DATE: 1AUG08	
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 MIC S 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.

DEXP8 - 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%.

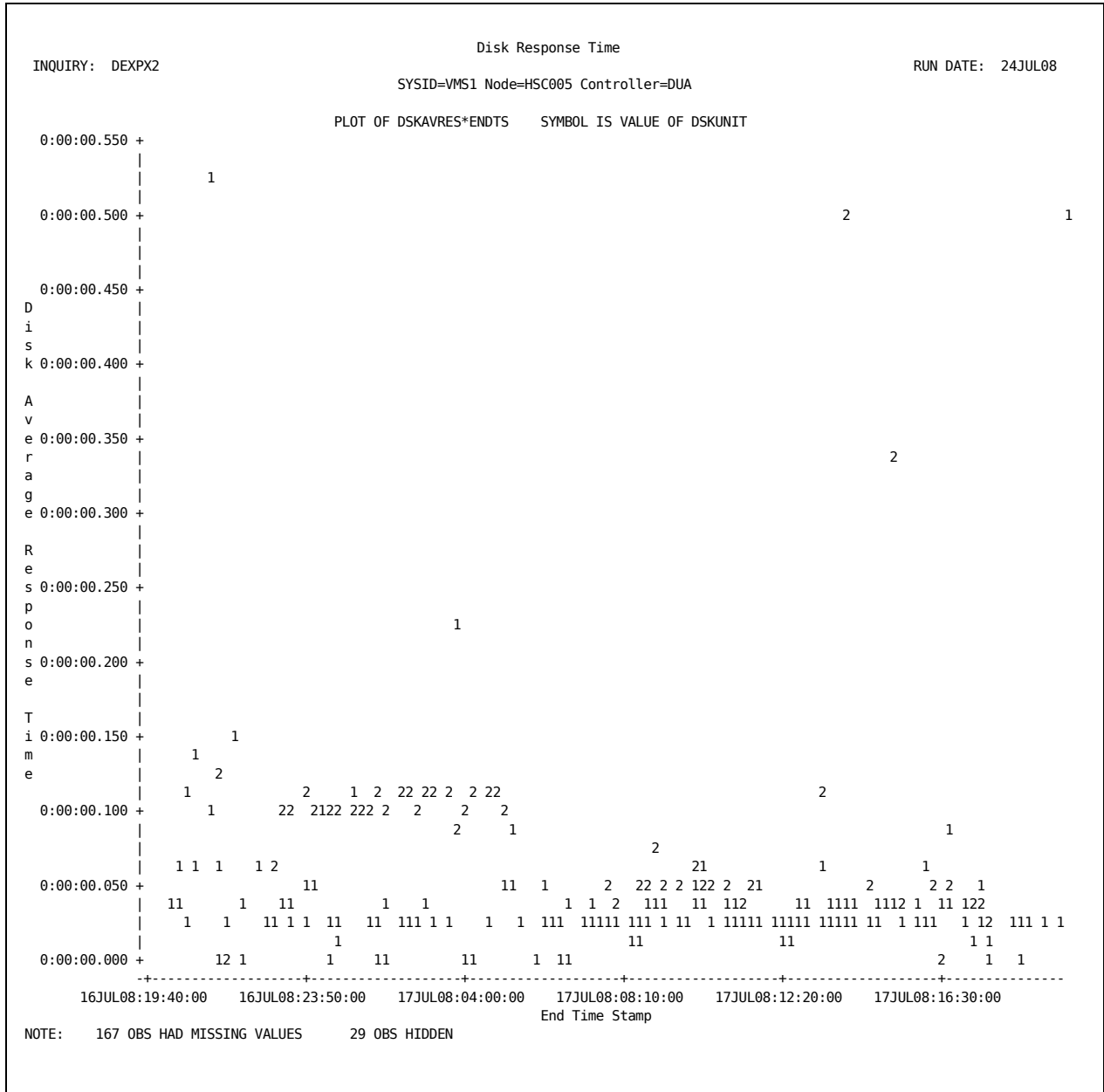


Figure 3-201. Disk Response Time

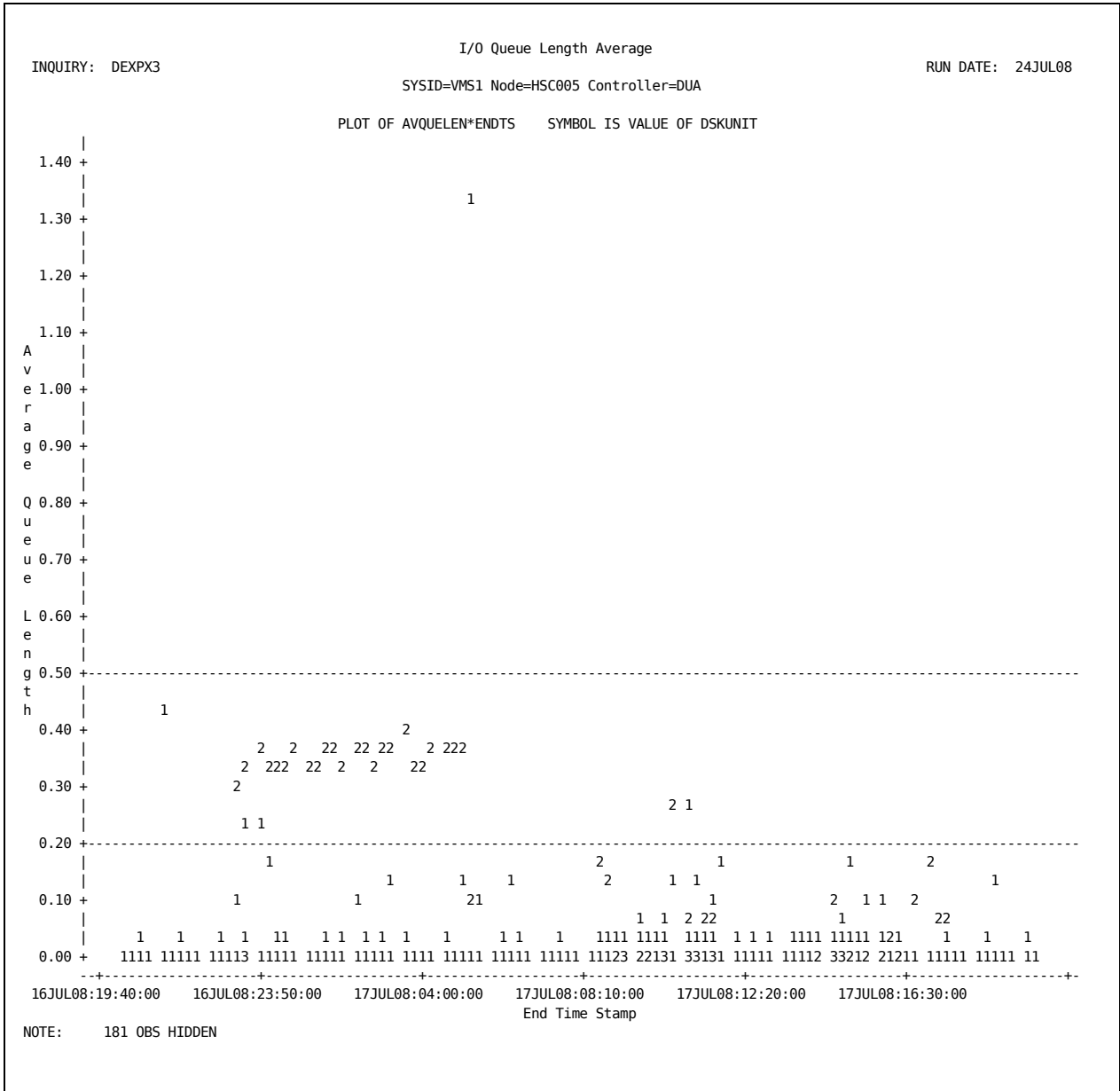


Figure 3-203. Service Performed by the System

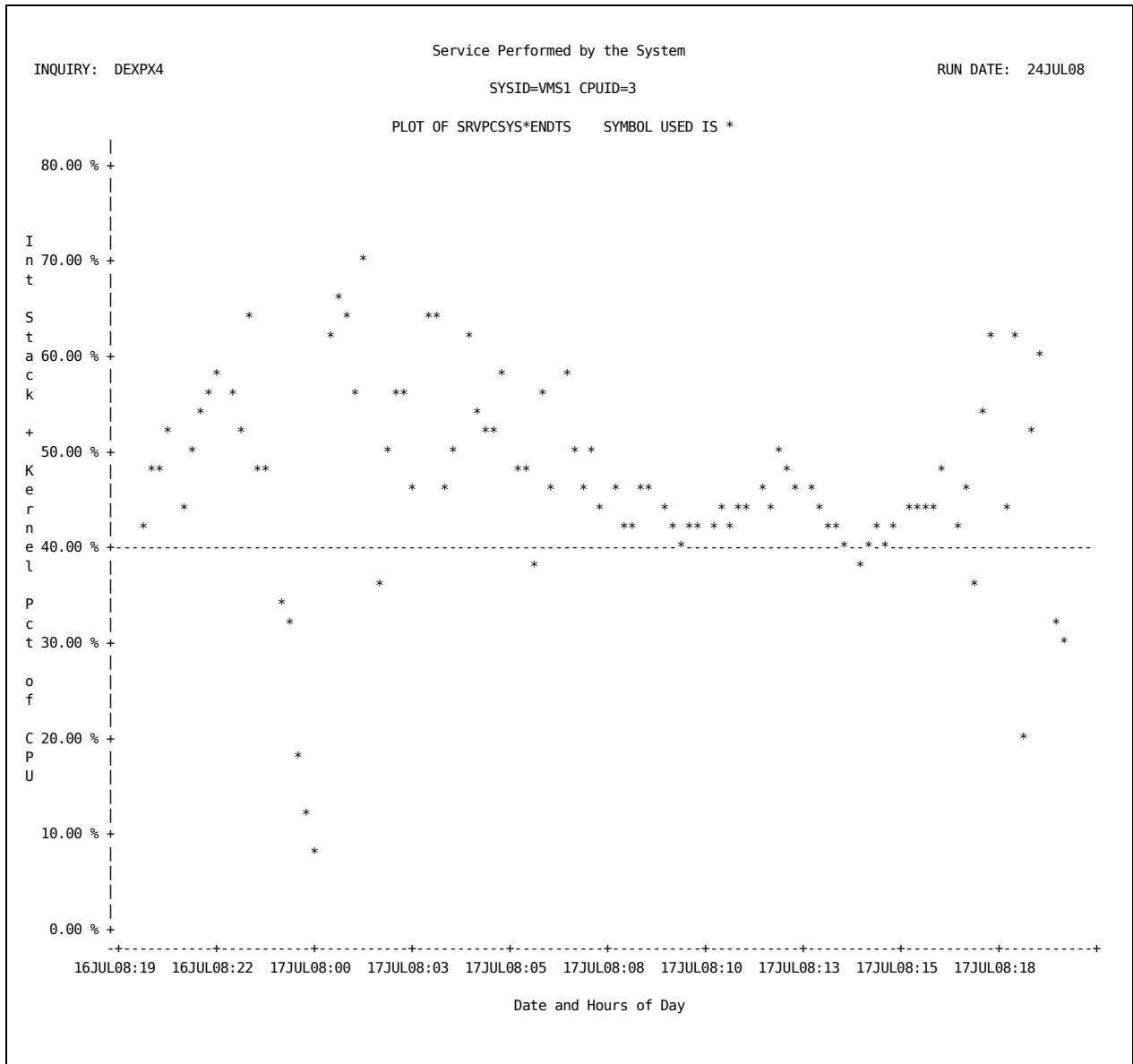


Figure 3-202. I/O Queue Length Average

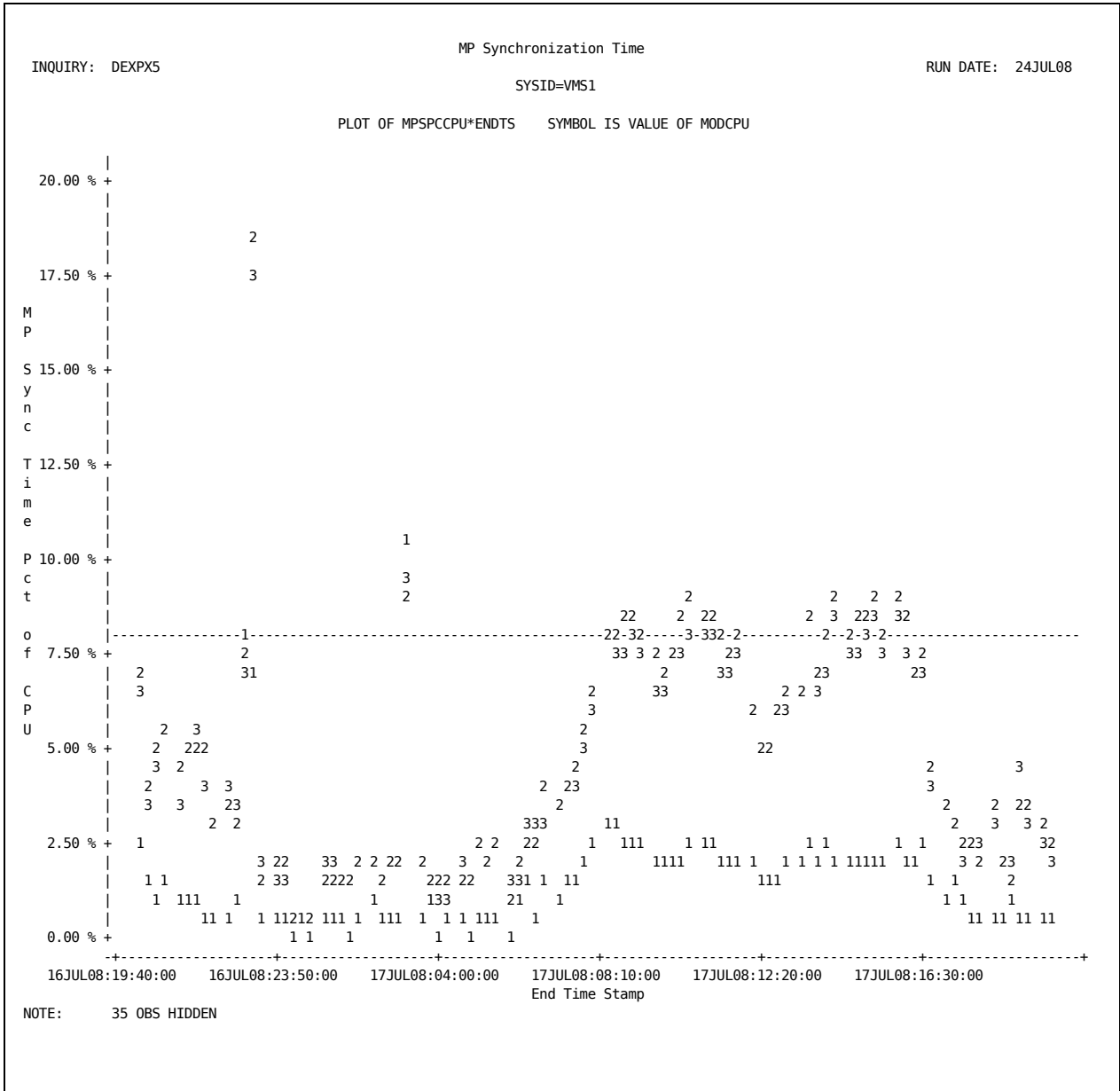


Figure 3-204. MP Synchronization Time

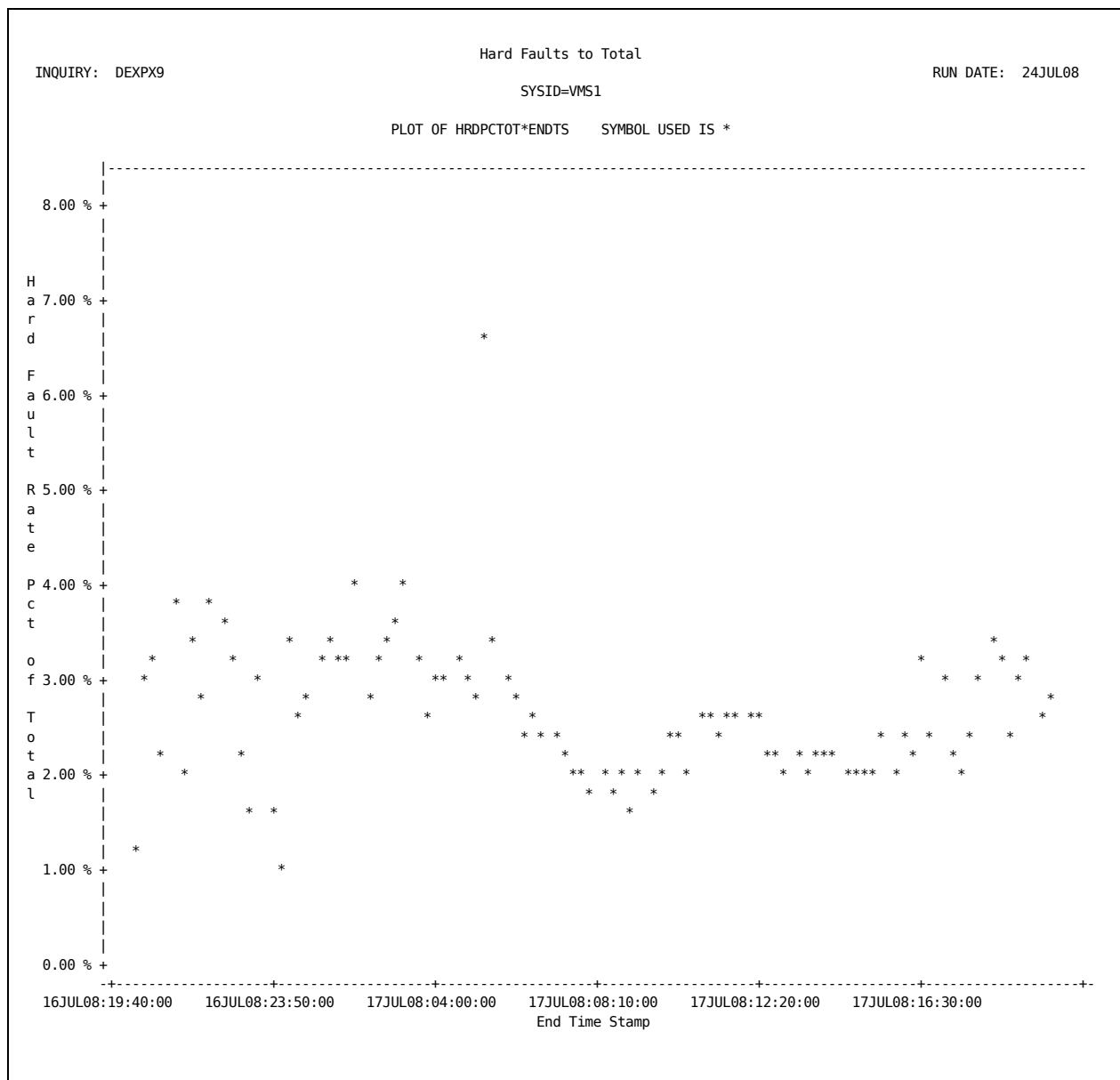


Figure 3-207. Hard Faults to Total

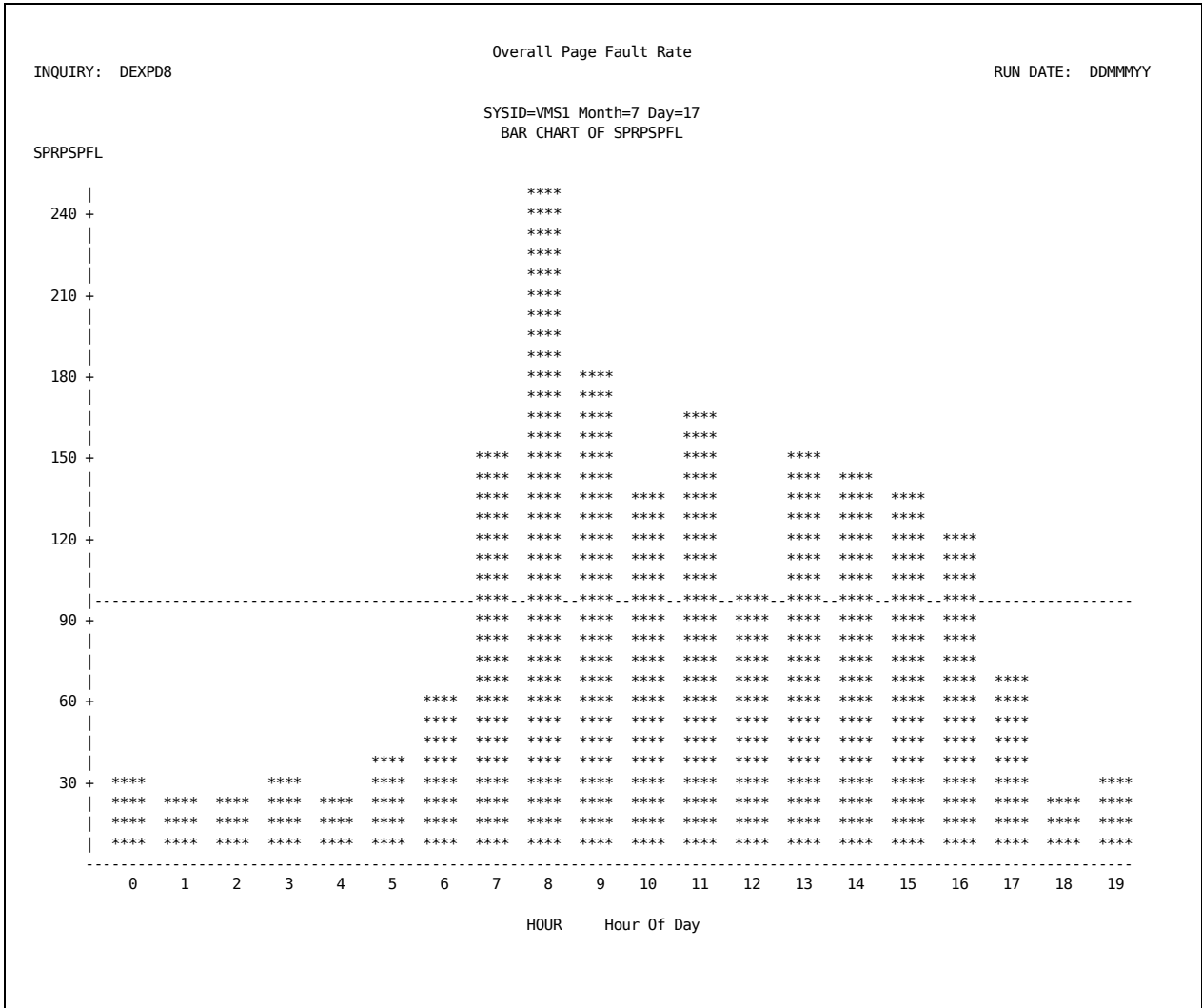


Figure 3-209. Overall Page Fault Rate

Daily System Status - Uptime & Memory Use										
INQUIRY: DEXLDH		System Identifier = VMS2 Hardware = VAX 8350				Year = 08 Month = 7		RUN DATE: 24JUL08		
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		SYSID VMS1										RUN DATE: 24JUL08	
Date	Hour	Avg CPU Percent	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			SYSID VMS2					RUN DATE: 24JUL08		
Date	Disk Device	Device Type	Blocks Total	Blocks Free	Pct of Space Used	Disk Avg Response Time	Disk Avg I/O 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

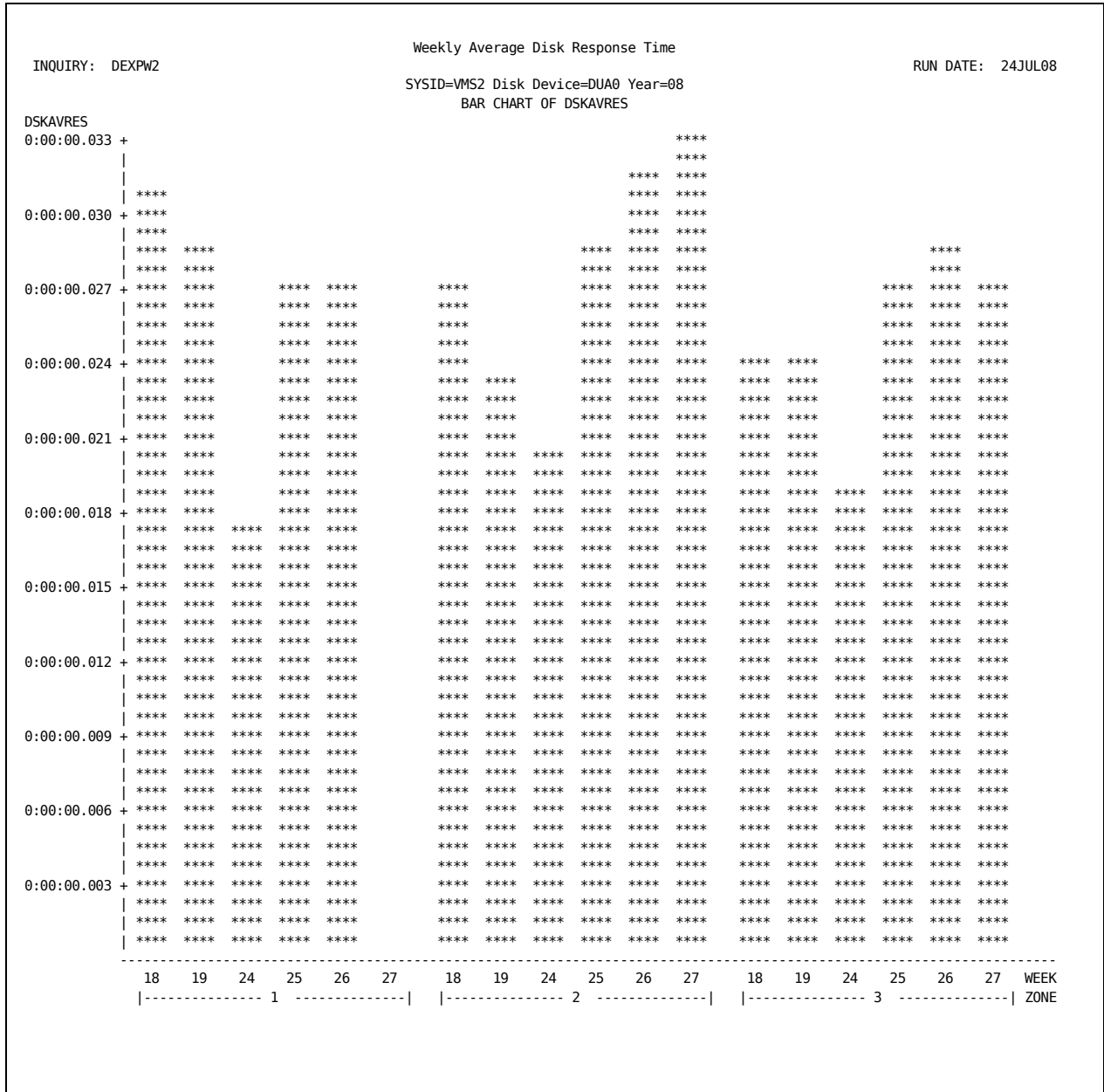


Figure 3-213. Weekly Average Disk Response Time

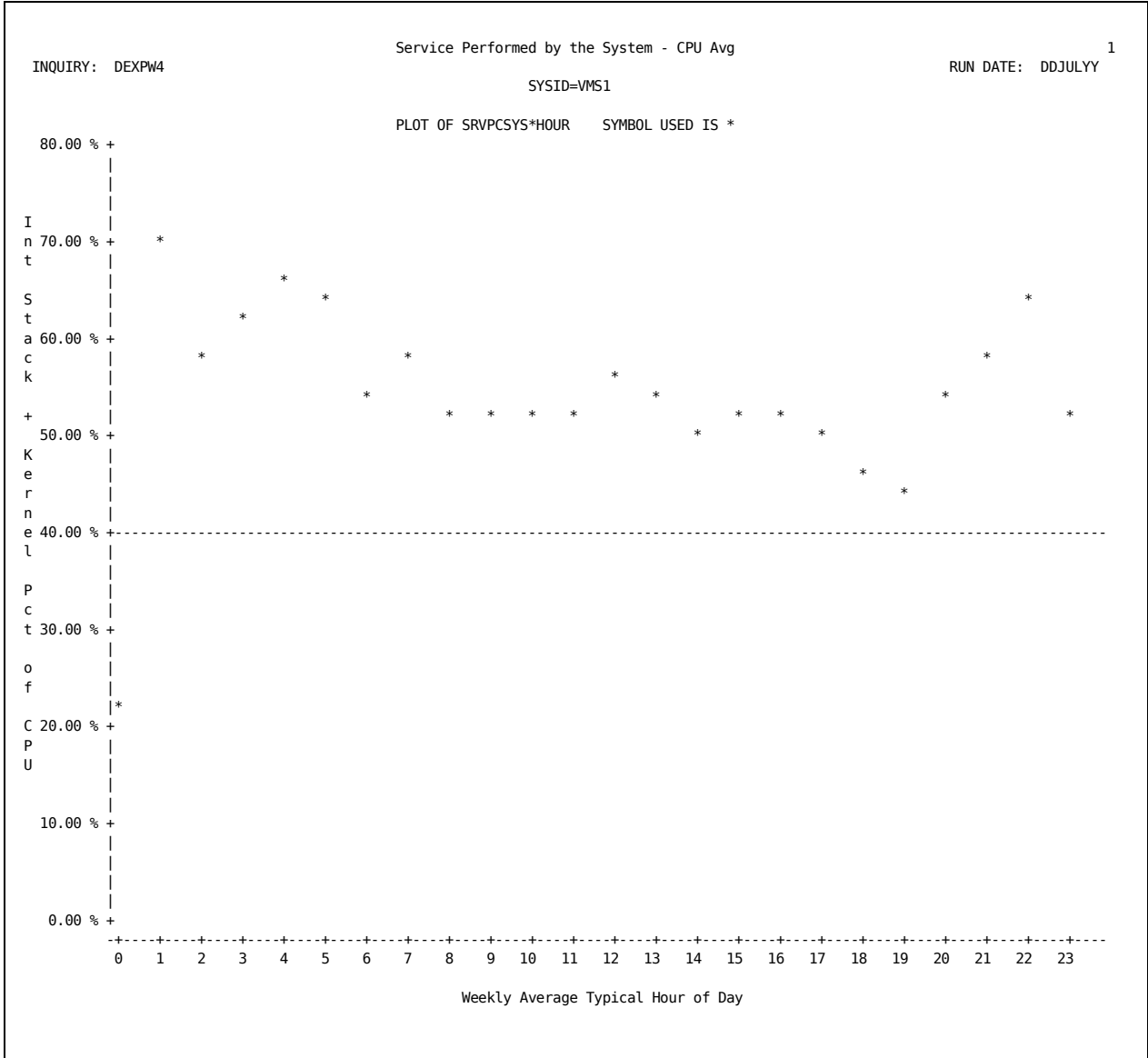


Figure 3-214. Service Performed by the System - CPU Avg

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 be 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		System Identifier = VMS2 Date = DDMMYY:00:00					RUN DATE: DDMMYY	
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
ACTING	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
BBBTAGL	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: DDMMYY

System Identifier = VMS2 Date = DDMMYY: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	ALLINI	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	MDIMP41	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	STXDEM00	DUA0	23	59	67	46.82 %	5	0.492
105	JVOLM	DUA1	10	90	15	85.71 %	11	0.410
99	AHZEDEL	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			System Identifier = VMS2 Date = 09JUL08:00:00			RUN DATE: 24JUL08		
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	ALLINI	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	STXDEM00	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	AHZEDEL	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	PRINTERS	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				System Identifier = VMS2 Date = DDMMYY:00:00 Device Name = DUA0			RUN DATE: DDMMYY
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: DDMMYY

System Identifier = VMS2 Date = DDMMYY: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>OAS\$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>OAHLP.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			System Identifier = VMS2 Date = DDMMYY:00:00		Device Name = DUA2	
			RUN DATE: DDMMYY			
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	
SEABTO,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			System Identifier = VMS2 Year = 08 Month = 7 Day = 9			RUN DATE: 24JUL08
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 %
	DBD8D	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 %
	FWVT	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	ALLINI	DUA0	220.090	7.289	96.79 %
160.043	OWDCL	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		System Identifier = VMS2 Year = 08 Month = 7 Day = 9			RUN DATE: 24JUL08	
	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	0WDFIN	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	STXDEM00	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	PRINTERS5	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

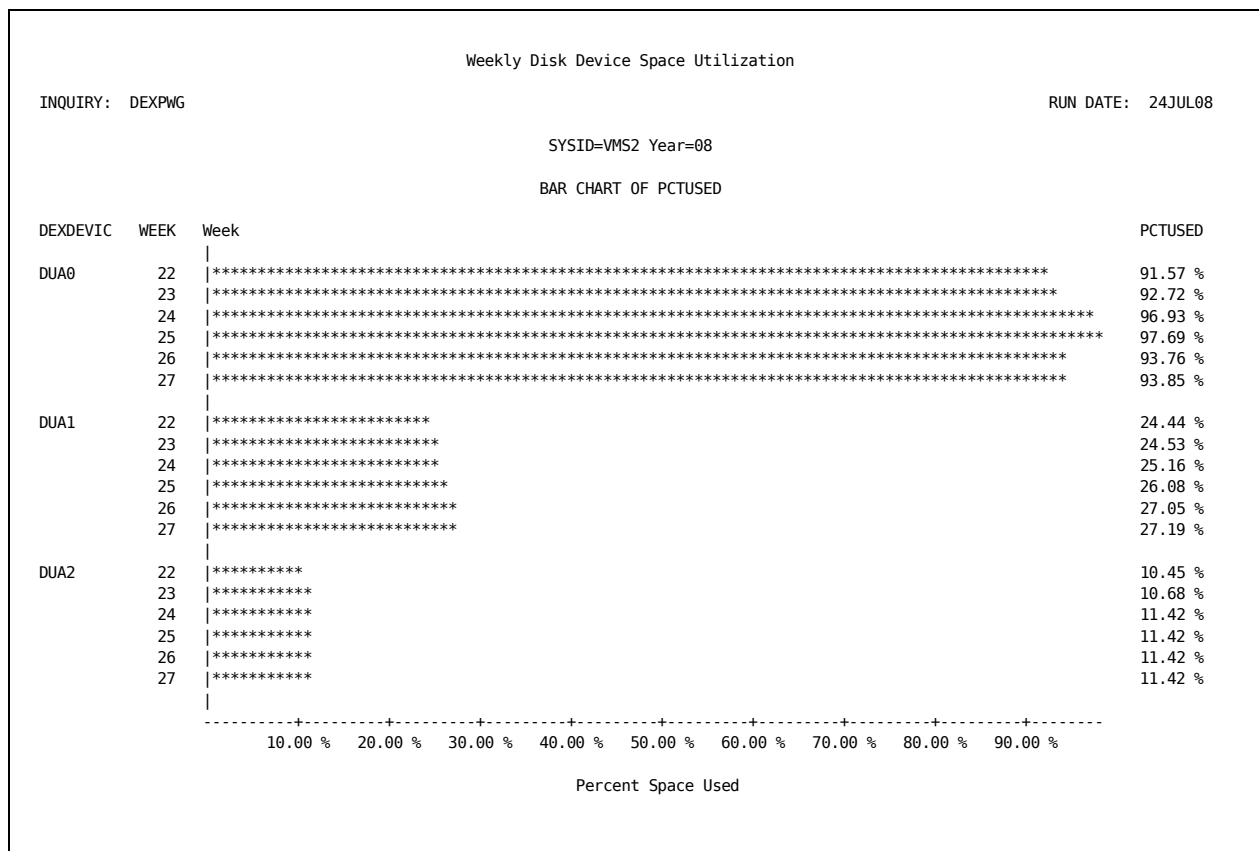


Figure 3-311. Weekly Disk Device Space Utilization

Monthly Disk Usage Summary							
INQUIRY: DEXLMA		System Identifier = VMS2 Year = 08 Month = 6				RUN DATE: 24JUL08	
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	FWVT	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		System Identifier = VMS2 Year = 08 Month = 6				RUN DATE: 24JUL08	
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 %	
		ALLINI	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	FWVT	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 %	
		SWIRLL0	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
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		System Identifier = VMS2 Year = 08 Month = 6				RUN DATE: 24JUL08	
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 %	
		FWVT	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
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

Chapter 4: EXCEPTIONS

This chapter is not yet available.

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		Date Generated:
Information Area		Tue, May 12, 2009
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		Date Generated:
Information Area		Tue, May 12, 2009
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		Date Generated:
Information Area		Tue, May 12, 2009
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 section contains the following topics:

- [5.1 Data Element Naming Conventions](#) (see page 79)
- [5.2 VAX/VMS Accounting Information Area Files](#) (see page 80)
- [5.3 VAX/VMS Monitor Information Area Files](#) (see page 107)

[5.4 VAX/VMS System Usage Information Area Files](#) (see page 159)

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: Tue, May 12, 2009

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: Tue, May 12, 2009

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 DEXPTY - Priority
 X....E DEXQUEUE - Queue Name
 X....E DEXRTID - 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.... MICSVER - CA MICS Version Number
 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. PR CFLTIO - 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 &pDEM.X..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 &pDEM.X..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: Tue, May 12, 2009

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: Tue, May 12, 2009

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 DEXPTY - 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.... MICSVER - CA MICS Version Number
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 IMGCCOST - 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.. IMGPAFGL - Page File Peak Usage
X..M.. IMGPAFGL - 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 lose 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
DEXPTY - 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 &pDEM..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 &pDEM..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: Tue, May 12, 2009

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: Tue, May 12, 2009

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 DEXACNT - 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 DEXPTY - 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.... MICSVER - CA MICS Version Number
 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. PR CFLTIO - Page Fault I/Os
 X..MY. PR CIMAGE - Image Count
 X..MY. PR CPAGFL - Page File Peak Usage
 X..MY. PR CPAGFT - Page Faults
 X..MY.E PR CVOLS - Volumes Mounted

Maximum Data Elements

X..MY. PR CWKSPK - 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: Tue, May 12, 2009

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: Tue, May 12, 2009

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 DEXPTY - 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.... MICSVER - CA MICS Version Number
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 PRQUETM - Queue Time
X..MY.E PRQSYMTM - Symbiont Time

Minimum Data Elements

X..MY.E PRQUETS - 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 &pDEM..DEAPRQ01;
IF DATEPART(ENDTS)=TODAY()-1;
PROC FREQ;
  TABLE DEXUSER;
```

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

```
DATA;
SET &pDEM..DEAPRQ01 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 DEAINTE 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	N/A					
YEARS	N/A					
TABLES	N/A					

Generation Date: Tue, May 12, 2009

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. DEAINTE Time-Span Granularity Chart

5.2.5.2 DEAINTE 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: Tue, May 12, 2009

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 DEXPTY - 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..... MICSVER - CA MICS Version Number
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 DEAIN 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 DEAINnn 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 DEAINnn 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 DEAINTE Retrieval Example

This section presents a typical DEAINTE retrieval example.

1. Print the times all systems were initialized yesterday.

```
DATA;  
SET &pDEMEX..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: Tue, May 12, 2009

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: Tue, May 12, 2009

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.... MICSVER - CA MICS Version Number
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: Tue, May 12, 2009

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: Tue, May 12, 2009

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..... MICSVER - CA MICS Version Number
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 &pDEM.X..DEMMOD01 END=EOF;
IF DATEPART(ENDTS)=TODAY()-1;
IF SYSID='DEV1';
RETAIN SUP SUPUSR 0;
SUP+MODSUPTM;
SUPUSR=SUPUSR+MODSUPTM+MODUSRMTM;
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 &pDEM.X..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: Tue, May 12, 2009

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: Tue, May 12, 2009

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..... MICSVER - CA MICS Version Number
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 DSKI00PS - Disk I/O Operations
XDWM..E DSKI0QUE - 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 \$122\$DRA2: 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 DSKI00PS;
```

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

```
DATA;
SET &pDEM..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: Tue, May 12, 2009

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: Tue, May 12, 2009

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.... MICSVER - CA MICS Version Number
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 SCMSGRC - Sequenced Messages Received
XDWM..E SCMSGSN - 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 &pDEM. .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 &pDEM..DEMCS01 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: Tue, May 12, 2009

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: Tue, May 12, 2009

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.... MICSVER - CA MICS Version Number
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

XDW.. RMSALAST - File Append Lock Blocking ASTs
XDW.. RMSALCNV - File Append Lock Conversions
XDW.. RMSALDEQ - File Append Lock Dequeues
XDW.. RMSALENQ - File Append Lock Enqueues
XDW.. RMSBUCKT - File Bucket Splits
XDW..E RMSCLOSE - File Closes
XDW..E RMSCONN - File Connects
XDW..E RMSDELET - File Deletes
XDW..E RMSDISCN - File Disconnects
XDW..E RMSEXTBL - File Extend Blocks
XDW..E RMSEXTEN - File Extends
XDW.. RMSFBAST - File Lock Blocking ASTs
XDW..E RMSFLUSH - File Flushes
XDW.. RMSGBAST - File Global Buffer Blocking ASTs
XDW.. RMSGBCNV - File Global Buffer Conversions
XDW.. RMSGBDEQ - File Global Buffer Dequeues
XDW.. RMSGBENQ - File Global Buffer Enqueues
XDW.. RMSGBRIO - File Global Buffer Read I/Os
XDW.. RMSGBWIO - File Global Buffer Write I/Os
XDW.. RMSGCATT - File Global Cache Attempts
XDW.. RMSGCHIT - File Global Cache Hits
XDW..E RMSGETBT - File GET Bytes
XDW.. RMSGSCNV - File Global Section Conversions
XDW.. RMSGSDEQ - File Global Section Dequeues
XDW.. RMSGSENG - File Global Section Enqueues
XDW..E RMSKEYFD - Keyed Finds
XDW..E RMSKEYGT - Keyed GETs
XDW..E RMSKEYPT - Keyed PUTs
XDW.. RMSLBAST - File Local Buffer Blocking ASTs
XDW.. RMSLBCNV - File Local Buffer Conversions
XDW.. RMSLBDEQ - File Local Buffer Dequeues
XDW.. RMSLBENQ - File Local Buffer Enqueues
XDW.. RMSLBRIO - File Local Buffer Read I/Os
XDW.. RMSLBWIO - File Local Buffer Write I/Os
XDW.. RMSLCATT - File Local Cache Attempts
XDW.. RMSLCHIT - File Local Cache Hits
XDW.. RMSLKCNV - File Lock Conversions
XDW.. RMSLKDEQ - File Lock Dequeues
XDW.. RMSLKENQ - File Lock Enqueues
XDW.. RMSMULTB - File Multibucket Splits
XDW..E RMSOPEN - File Opens
XDW..E RMSPUTBT - File PUT Bytes
XDW..E RMSRDBT - File Read Bytes
XDW..E RMSREAD - File Reads
XDW..E RMSREWIND - File Rewinds
XDW..E RMSRFAFD - Record File Address Finds
XDW..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 &pDEM.X..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 &pDEM.X..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	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					

Generation Date: Tue, May 12, 2009

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: Tue, May 12, 2009

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..... MICSVER - CA MICS Version Number
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 SPRPCNT - 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 SPRENQNG - 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)

XDW..E SPRTRNCT - TRN TRANS Record Count (Type 22)
XDW..E SPRTRNDR - TRN TRANS Record Duration (Type 22)
XDW.. SPRTURN - Window Turns
XDW.. SPRUNI - Unlocks, Incoming
XDW.. SPRUNL - Unlocks, Local
XDW.. SPRUNO - Unlocks, Outgoing
XDW.. SPRVLOK - Volume Lock Waits
XDW..E SPRWIPFL - Write In Progress Page Faults
XDW.. SPRWTURN - Window Turns

Minimum Data Elements

XDW.. SPRMNF - Minimum Free Page Count
XDW..E SPRMNFRE - Minimum Free Page List Size
XDW.. SPRMNHOL - Minimum Unused KBytes
XDW..E SPRMNIFP - IO Minimum Free Page Count
XDW..E SPRMNIMP - IO Minimum Modified Page Count
XDW..E SPRMNIPF - Minimum Intermediate Req Packets Free
XDW.. SPRMNLBK - Minimum Largest Block
XDW..E SPRMNLPF - Minimum Large Request Packets Free
XDW..E SPRMNLPK - Minimum Large Request Packets Left
XDW.. SPRMNMP - Minimum Modified Page Count
XDW..E SPRMNPFP - PGE Minimum Free Page Count
XDW..E SPRMNPMP - PGE Minimum Modified Page Count
XDW.. SPRMNSBK - Minimum Smallest Block
XDW..E SPRMNSPF - Minimum Small Request Packets Free
XDW.. SPRMNUCN - Minimum Unused Contiguous Space

Maximum Data Elements

XDW.. SPRMSTAB - Max Transaction Abort Rate
XDW.. SPRMSTAD - Max Transaction Remote Add Rate
XDW.. SPRMSTBR - Max Transaction Remote Start Rate
XDW.. SPRMSTCO - Max Transaction Total Commit Rate
XDW.. SPRMSTEN - Max Transaction End Rate
XDW.. SPRMSTOP - Max Transaction One-Phase Init Rate
XDW.. SPRMSTPR - Max Transaction Prepare Rate
XDW.. SPRMSTST - Max Transaction Start Rate
XDW.. SPRMXCEF - Maximum Common Event Flag Wait
XDW..E SPRMXCLK - Maximum Current Locks
XDW..E SPRMXCP - Maximum Current Processes
XDW.. SPRMXCPW - Maximum Collided Page Wait
XDW..E SPRMXCRS - Maximum Current Resources
XDW..E SPRMXCSI - Maximum Compute State Wait, Inswapped
XDW..E SPRMXCSO - Maximum Compute State Wait, Outswapped
XDW.. SPRMXFPW - Maximum Free Page Wait
XDW.. SPRMXHWI - Maximum Hibernate Wait, Inswapped
XDW.. SPRMXHWO - Maximum Hibernate Wait, Outswapped
XDW..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.. SPRPS SPL - 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 LEF0, SUSP0, and COM0 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 &pDEMXX..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 &pDEMXX..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 &pDEMXX..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 &pDEMXX..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 &pDEMXX..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 &pDEMXX..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 &pDEMXX..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 &pDEMXX..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 &pDEMXX..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 &pDEMXX..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 &pDEMXX..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: Tue, May 12, 2009

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: Tue, May 12, 2009

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.... MICSVER - CA MICS Version Number
 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: Tue, May 12, 2009

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: Tue, May 12, 2009

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.... MICSVER - CA MICS Version Number
 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: Tue, May 12, 2009

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: Tue, May 12, 2009

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.... MICSVER - CA MICS Version Number
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 DESDKU 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 DESDKU 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 DKDIRTL 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: Tue, May 12, 2009

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: Tue, May 12, 2009

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.... MICSVER - CA MICS Version Number
X....E ORGSYSID - Originating System Identification
XDWMY.E STARTTS - Start Time Stamp

Retained Data Elements

X.... PRXEVTNF - 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
PRXEVRTF - 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: Tue, May 12, 2009

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: Tue, May 12, 2009

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.... MICSVER - CA MICS Version Number
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 SYUOPTIM - 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;
```

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 section contains the following topics:

[6.1 Data Collection and Transport to CA MICS](#) (see page 187)

[6.2 VMS ACCOUNTING Record Descriptions](#) (see page 202)

[6.3 VMS MONITOR Record Descriptions](#) (see page 207)

[6.4 CA MICS VMS System Usage \(DEXSUS\) Record Descriptions](#) (see page 214)

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

DEXSUS01 - 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 DEXSUS01 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.
 - o 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.
 - o 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.
 - o 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.
 - o 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
 - o 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:

- 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 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:

- 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.
- o 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:

- 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 DEINT 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()
```


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 CA MICS Planning, Installation, Operation, and Maintenance (PIOM) Guide.

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 section contains the following topics:

- [7.1 Environmental Considerations](#) (see page 222)
- [7.2 Complex Level Parameters](#) (see page 222)
- [7.3 Unit Level Parameters](#) (see page 233)
- [7.4 Other Related Parameters](#) (see page 284)

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..
+-----+		

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 D0;  
|  
| * 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.. |  
+-----+
```

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 D0;  
|  
| * BUILD ACCOUNT CODE FIELDS AS IN THE WORKSHEET 7-1  
|  
|   DEXACT1=field source 1 ;  
|   ...  
|   ...  
|   ...  
|   DEXACTn=field source n ;  
|  
| -----  
| -----  
| -----  
| -----  
|  
| END;  
| ELSE D0;  
|  
| * 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.. |  
+-----+
```

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(JCLGEN) or by *
* specifying DAILY in prefix.MICS.PARMS(JCLGEN) *
* and executing prefix.MICS.CNTL(JCLGEN). *
*
* 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(JCLGEN) or by specifying
*       DAILY in prefix.MICS.PARMS(JCLGEN) and
*       executing prefix.MICS.CNTL(JCLGEN).
*
*       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 (or IEHPRGM) 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) DEXSUS _____(orgsysid) _____(ddname) _____(default 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_parameters MULTWORK NOMULT __ fff ... fff RESTART _____ YES/NO
...5...10...15...20...25...30...35...40...45...50...55...60...65...70..	

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.

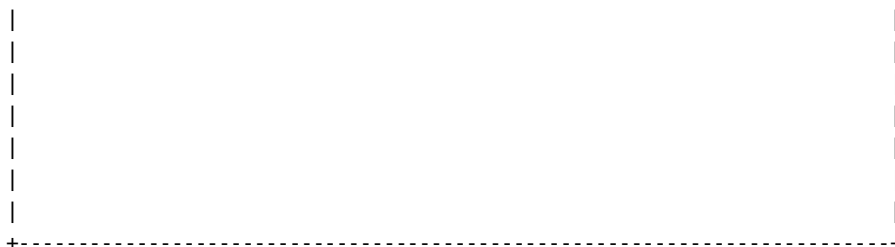
Providing Data to CA MICS

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=_____ |

```



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).

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 is 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 can be specified whethert 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, 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 |
|                                                                           |
```

```

| PARS Library Member is DBMODEL
| Reference Sections: 7.3.4.2
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |      |
| 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) |
|DEMRRM|__ (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.. |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| INSTALLATION PREPARATION WORKSHEET: Database Data Retention Definitions |
| PARS Library Member is DBMODEL
| Reference Sections: 7.3.4.2
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |      |
| 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.. |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

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).



# 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 CA MICS Planning, Installation, Operation, and Maintenance (PIOM) Guide.

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 section contains the following topics:

[8.1 Installing the DEXSUS.COM Collection Routine](#) (see page 289)

## 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.

# 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.

This section contains the following topics:

[9.1 Processing Considerations](#) (see page 291)

[9.2 Daily Update Processing Flow](#) (see page 292)

[9.3 User Exits](#) (see page 296)

[9.4 Using COMMON Input DDnames and DEXINPUT exits](#) (see page 299)

## 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 |      | VAX/VMS Accounting Files | VAX/VMS System Usage Files |
|------------------------|------|--------------------------|----------------------------|
| Name                   | Type |                          |                            |
| -----                  | ---- | -----                    | -----                      |
| 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 |      | VAX/VMS Monitor Files | VAX/VMS System Usage Files |
|---------------------|------|-----------------------|----------------------------|
| Name                | Type |                       |                            |
| -----               | ---- | -----                 | -----                      |
| 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                     |

| CA MICS DEXSUS Records    |      | VAX/VMS     |              |
|---------------------------|------|-------------|--------------|
| Name                      | Type |             | System Usage |
| ----                      | ---- |             | -----        |
| 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       |

#### 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, DEXSUB, 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    | USRINT    |
| DEA_LF    | USR_LF    |
| DEAPRQ    | USRSPRQ   |
| DEMPRO    | USRSPRO   |
| DEMSTA    | USRSTA    |
| DEMOD     | USRSMOD   |
| DEMPGE    | USRSPGE   |
| DEMIOV    | USRIOV    |
| DEMFCP    | USRFCP    |
| DEMPOL    | USRSPOL   |
| DEMLOK    | USRLOK    |
| DEMDNT    | USRSDNT   |
| DEMFSC    | USRFSFC   |
| DEMDSK    | USRSDSK   |
| DEMDLK    | USRSDLK   |
| DEMSCS    | USRSSCS   |
| DEMSYT    | URSSYT    |
| DEMCLS    | USRCLS    |
| DEMRMS    | USRMRMS   |
| DEMSC     | USRMSC    |
| DEMSPR    | URSSPR    |
| DEMTRN    | USRSTRN   |
| DEMVPR    | URSVPR    |
| DESDKD    | USRSDKD   |
| DESDKQ    | USRSDKQ   |
| DESDKU    | USRDKU    |
| DESPRX    | URSPRX    |
| DESSYU    | URSSYU    |

### 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   |

(\*) 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_CHK;
%MEND DEXINPTM;

%MACRO DEXINPTA;
 ROFFSET=14;
 INPUT @1
 VAXID $CHAR10. VAXDS $CHAR4.
 @ ;
 IF VAXDS NE "ACCO" THEN GOTO EOF_CHK;
%MEND DEXINPTA;

%MACRO DEXINPTS;
 ROFFSET=14;
 INPUT @1
 VAXID $CHAR10. VAXDS $CHAR4.
 @ ;
 IF VAXDS NE "DXSU" THEN GOTO EOF_CHK;
%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
```

# 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.

This section contains the following topics:

[10.1 Parameter Modification](#) (see page 305)

## 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.

# Appendix A: MESSAGES

---

## 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 PARM(S) 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.PARM(S). 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 PARM(S).

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



# Appendix B: DATA DICTIONARY

---

The Data Dictionary is only available at your site, where it has been customized to your configuration and your product change level.

To see the Data Dictionary at your site, follow the instructions under Document Browse in the Document Access guide.