VMS

VMS Monitor Utility Manual

digital

VMS Monitor Utility Manual

Order Number: AA–LA45A–TE

April 1988

This document describes how to use the Monitor Utility on VMS operating systems.

Revision/Update Information: This document supersedes the *VAX/VMS Monitor Utility Reference Manual*, Version 4.4.

Software Version: VMS Version 5.0

April 1988

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	UNIBUS
DEC/CMS	EduSystem	VAX
DEC/MMS	IAS	VAXcluster
DECnet	MASSBUS	VMS
DECsystem–10	PDP	VT
DECSYSTEM–20	PDT	
DECUS	RSTS	
DECwriter	RSX	**digital** ™

ZK4551

HOW TO ORDER ADDITIONAL DOCUMENTATION
DIRECT MAIL ORDERS

Production Note

This book was produced with the VAX DOCUMENT electronic publishing system, a software tool developed and sold by DIGITAL. In this system, writers use an ASCII text editor to create source files containing text and English-like code; this code labels the structural elements of the document, such as chapters, paragraphs, and tables. The VAX DOCUMENT software, which runs on the VMS operating system, interprets the code to format the text, generate a table of contents and index, and paginate the entire document. Writers can print the document on the terminal or line printer, or they can use DIGITAL-supported devices, such as the LN03 laser printer and PostScript™ printers (PrintServer 40 or LN03R ScriptPrinter), to produce a typeset-quality copy containing integrated graphics.

™ PostScript is a trademark of Adobe Systems, Inc.

Contents

Contents

INDEX

Contents

TABLES

Preface

Intended Audience

This manual is intended for VMS system managers, operators, and system programmers.

Document Structure

This document consists of the following six sections:

- Description—Provides a full description of the Monitor Utility (MONITOR).

- Usage Summary—Outlines the following MONITOR information:

  - Invoking the utility
  - Exiting from the utility
  - Directing output
  - Restrictions or privileges required

- Qualifiers—Describes MONITOR qualifiers, including format, parameters, and examples.

- Commands—Describes MONITOR commands, including format, parameters, and examples.

- Examples—Provides additional MONITOR examples.

- Appendix A—Provides supplemental MONITOR information.

Associated Documents

For additional information on the topics covered in this document, refer to the *VMS DCL Dictionary* and the *Guide to VMS Performance Management*.

Preface

Conventions

Convention	Meaning
RET	In examples, a key name (usually abbreviated) shown within a box indicates that you press a key on the keyboard; in text, a key name is not enclosed in a box. In this example, the key is the RETURN key. (Note that the RETURN key is not usually shown in syntax statements or in all examples; however, assume that you must press the RETURN key after entering a command or responding to a prompt.)
CTRL/C	A key combination, shown in uppercase with a slash separating two key names, indicates that you hold down the first key while you press the second key. For example, the key combination CTRL/C indicates that you hold down the key labeled CTRL while you press the key labeled C. In examples, a key combination is enclosed in a box.
$ SHOW TIME 05-JUN-1988 11:55:22	In examples, system output (what the system displays) is shown in black. User input (what you enter) is shown in red.
$ TYPE MYFILE.DAT . . .	In examples, a vertical series of periods, or ellipsis, means either that not all the data that the system would display in response to a command is shown or that not all the data a user would enter is shown.
input-file, . . .	In examples, a horizontal ellipsis indicates that additional parameters, values, or other information can be entered, that preceding items can be repeated one or more times, or that optional arguments in a statement have been omitted.
[logical-name]	Brackets indicate that the enclosed item is optional. (Brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
quotation marks apostrophes	The term quotation marks is used to refer to double quotation marks ("). The term apostrophe (') is used to refer to a single quotation mark.

New and Changed Features

Version 5.0 of the Monitor Utility includes the following new functions:

- A new MONITOR MSCP_SERVER command that produces statistics useful in tuning an MSCP server.

- A new MONITOR RMS command that produces a variety of statistics related to VMS Record Management Services.

- The MONITOR MODES command now produces statistics that pertain to symmetric multiprocessor systems.

- The MONITOR IO command now produces data for split transfers.

MONITOR Description

The Monitor Utility (MONITOR) is a system management tool that enables you to obtain information on operating system performance. Using MONITOR, you can monitor classes of systemwide performance data (such as system I/O statistics, page management statistics, and time spent in each of the processor modes) at specifiable intervals, and produce several types of output.

To monitor a particular class of information, you specify the class name corresponding to the information class on the MONITOR command line. For example, to monitor page management statistics, specify the PAGE class name in the MONITOR command. MONITOR collects system performance data by class and produces the following three forms of optional output:

- A disk recording file in binary format

- Statistical terminal displays

- A disk file containing statistical summary information in ASCII format

The utility initiates a single MONITOR request for the classes of performance data specified each time you enter a command in the following form:

MONITOR [/qualifier[,...]] classname[,...] [/qualifier[,...]]

Regardless of the order in which you specify class-name parameters, MONITOR always executes requests in the following sequence:

    PROCESSES
    STATES
    MODES
    PAGE
    IO
    FCP
    POOL
    LOCK
    DECNET
    FILE_SYSTEM_CACHE
    DISK
    DLOCK
    SCS
    SYSTEM
    CLUSTER
    RMS
    MSCP_SERVER

Depending on the command qualifiers specified, MONITOR collects system performance data from the running system or plays back data recorded previously in a recording file. When you play back data, you can display it, summarize it, and even rerecord it to reduce the amount of data in the recording file. The Examples section illustrates these operations in greater detail.

For additional information about interpreting the information the Monitor Utility provides, see the *Guide to VMS Performance Management*.

MONITOR Description

1   Class Types

Each MONITOR class consists of data items that, taken together, provide a statistical measure of a particular system performance category. The data items defined for individual classes are listed in the description of the MONITOR command in the Commands section.

There are two MONITOR class types, differentiated by the scope of the data items collected:

*   *System Classes,* in which the data items provide statistics on resource utilization for the entire system (CLUSTER, DECNET, DLOCK, FCP, FILE_SYSTEM_CACHE, IO, LOCK, MSCP_SERVER, PAGE, POOL, STATES, SYSTEM).

*   *Component Classes,* in which the data items provide statistics on the contribution of individual components to the overall system or cluster. These classes are DISK, MODES, PROCESSES, RMS (Record Management Services), and SCS (System Communication Services).

As an example of the distinction between MONITOR class types, the IO class includes a data item to measure all direct I/O operations for the entire system and is therefore a system class. The DISK class measures direct I/O operations for individual disks, and is therefore a component class.

2   Class-Name Qualifiers

The class-name qualifiers control the type of display and summary output format generated for each class name specified. They have no effect on the recording of binary data. Each of these qualifiers applies only to the immediately preceding class name. Class-name qualifiers must not appear as part of the command verb. Table MON–1 summarizes class-name qualifiers and defaults.

Table MON–1   MONITOR Class-Name Qualifiers

Class Name	Qualifiers	Defaults
ALL_CLASSES	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	See Commands section.
CLUSTER	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/CURRENT
DECNET	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
DISK	/ALL /AVERAGE /CURRENT /ITEM /MAXIMUM /MINIMUM /[NO]PERCENT	/ALL /ITEM=OPERATION_ RATE /NOPERCENT
DLOCK	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
FCP	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
FILE_SYSTEM_CACHE	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL

Table MON-1 (Cont.)   MONITOR Class-Name Qualifiers

Class Name	Qualifiers	Defaults
IO	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
LOCK	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
MODES	/ALL /AVERAGE /[NO]CPU /CURRENT /MAXIMUM /MINIMUM /[NO]PERCENT	/NOCPU /CURRENT /NOPERCENT
MSCP_SERVER	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
PAGE	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
POOL	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/ALL
PROCESSES	/TOPBIO /TOPCPU /TOPDIO /TOPFAULT	None
RMS	/ALL /AVERAGE /CURRENT /FILE /ITEM /MAXIMUM /MINIMUM	/ITEM=OPERATIONS /ALL
SCS	/ALL /AVERAGE /CURRENT /ITEM /MAXIMUM /MINIMUM /[NO]PERCENT	/ALL /ITEM=KB_MAP /NOPERCENT
STATES	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM /[NO]PERCENT	/CURRENT /NOPERCENT
SYSTEM	/ALL /AVERAGE /CURRENT /MAXIMUM /MINIMUM	/CURRENT

Following are the three categories of class-name qualifiers:

* Statistics qualifiers (/ALL, /AVERAGE, /CURRENT, /MAXIMUM, and /MINIMUM) specify which statistics appear in display and summary output. These are conflicting qualifiers; specify no more than one of these qualifiers with each class name in a MONITOR request. Statistics qualifiers cannot be used with the PROCESSES class name or for multifile summaries.

* The data transformation qualifier (/[NO]PERCENT) controls whether data for the selected class name is expressed as percentages of a whole. This qualifier can be used only with the STATES, DISK, MODES, and SCS class names, and is not allowed for multifile summaries.

* Class-specific qualifiers (/CPU, /ITEM, /FILE, /TOPBIO, /TOPCPU, /TOPDIO, and /TOPFAULT) control the output of a specific class.

  - /CPU is used with the MODES class name to produce information for specific CPUs in a multiprocessor configuration.

  - /ITEM is used with the component statistics class names DISK, RMS, and SCS to specify one or more data items for inclusion in display or summary output.

  - /FILE is used with the RMS class name to specify the RMS file to which a MONITOR RMS command applies.

MONITOR Description

> – /TOP is used with the PROCESSES class name to produce bar
> graphs showing the top processes instead of the standard summary
> and display output. Top processes are the heaviest consumers of the
> resource being monitored. Up to eight processes can be shown in
> each display. Note that the /TOP qualifiers are mutually exclusive.
> Specify no more than one in a single request.

3 Output Types

MONITOR can produce any combination of three forms of output for any
single MONITOR request. The forms are display output, recording file
output, and summary output. Output forms are specified with the /DISPLAY,
/RECORD, and /SUMMARY qualifiers, as follows:

- /DISPLAY produces output in the form of ASCII screen images. Screen
  images are written at a frequency governed by the /VIEWING_TIME
  qualifier.

- /RECORD produces a binary recording file containing data collected for
  requested classes; one record for each class is written per interval.

- /SUMMARY produces an ASCII file containing summary statistics for all
  requested classes over the duration of the MONITOR request.

If you specify /INPUT with any of these qualifiers, MONITOR collects
performance data from one or more previously created recording files;
otherwise, data is collected from counters and data structures on the running
system. The MONITOR request begins and ends at times specified by the
/BEGINNING and /ENDING qualifiers respectively.

3.1 Display Output

Display output consists of a series of terminal screen images. One screen
image for each requested class for each requested viewing interval is
produced. You can use any terminal supported by VMS with dimensions of
at least 80 columns by 24 rows. (You might have to enter the DCL command
SET TERMINAL to set the proper dimensions.) Display output can also be
routed to a file for subsequent printing.

The amount of time between screen displays is determined by the
/VIEWING_TIME value. Effective viewing time varies, however, depending
on whether you are running MONITOR on your local system or on a remote
node. (*Remote* in this context refers to use of the SET HOST command to
access another node.) For remote access, the time required to display the
screen is included in the viewing time, while for local access this time is not
included. Therefore, use a larger viewing time than the 3-second default
when running MONITOR on a remote system. The value appropriate for
remote access depends on your terminal baud rate. For a 9600–baud terminal
line, 6 seconds is a reasonable viewing time. For lower-speed lines, increase
the viewing time appropriately.

By pressing CTRL/W, you can temporarily override the /VIEWING_TIME
value and generate a new display immediately following the current one.
This feature is useful when the MONITOR display area has been overwritten
by an operator message. You can also use CTRL/W in conjunction with a
large /VIEWING_TIME value to generate display events on demand.

3.2   Display Data

All displayable data items are rates or levels except in the PROCESSES class. Rates are shown in number of occurrences per second. A level is a value that indicates the size of the monitored data item.

MONITOR can display any of four different statistics for each data item, as follows:

• Current rate or level

• Average rate or level

• Minimum rate or level

• Maximum rate or level

Average, minimum, and maximum statistics are measured from the beginning of the MONITOR request. The current statistic is the most recently collected value for the rate or level. Any or all of the statistics can be requested. For the DISK, MODES, SCS, and STATES classes, all statistics can be expressed as percentages.

3.3   Screen Formats

There are two basic screen formats used for displaying MONITOR class data: the single-statistic screen and the multiple-statistic screen. The formats vary slightly depending on whether the class being displayed is a system or component class.

The following three characteristics occur in both screen formats:

• The date and time appearing in the heading of each screen refer to the time the displayed data was originally collected.

• The name of the node on which the data was originally collected also appears in the heading (except when playing back files that do not contain node name information or when displaying CLUSTER class data). The node name is obtained from the SCSNODE system parameter or, if SCSNODE is null, from the SYS$NODE logical name established by DECnet.

• The bottom line of the display is used for status information about the current MONITOR request.

    – If data collection is from a file of previously recorded monitor data, the word PLAYBACK appears at the left margin of the line. If the currently running system is being monitored, the word does not appear.

    – If a summary file has been requested, the word SUMMARIZING appears in the middle of the line. If not, it does not appear.

    – If creation of a recording file has been requested, the word RECORDING appears at the right margin of the line. If not, it does not appear.

The PROCESSES, SYSTEM, AND CLUSTER classes have unique screen formats.

MONITOR Description

3.3.1 Single-Statistic Screen for System Classes

This bar-graph style screen is used whenever one statistic (current, average, minimum, or maximum) is requested. Example MON–1 shows the maximum statistic for the STATES class. For other classes and statistics, the screen format remains the same with different heading and data item descriptions. If the display of percentages is requested, the percent symbol (%) appears in the title and next to the numbers along the top of the graph. All values in this screen format are rounded up or down to seven whole numbers (except percentages, which are rounded to three whole numbers).

Example MON–1 Single-Statistic Screen

```
                              VAX/VMS Monitor Utility
                                  PROCESS STATES
                 +-----+            on node SAMPLE
                 | MAX |         31-DEC-1988 16:09:53
                 +-----+
                                 0         10        20        30        40
                                 + - - - - + - - - - + - - - - + - - - - -+
Collided Page Wait               1 |*
Mutex & Misc Resource Wait       3 |***
Common Event Flag Wait             |
Page Fault Wait                  2 |**
Local Event Flag Wait           28 |***************************
Local Evt Flg (Outswapped)       4 |****
Hibernate                       11 |**********
                                   |
Hibernate (Outswapped)           2 |**
Suspended                          |
Suspended (Outswapped)             |
Free Page Wait                     |
Compute                          4 |****
Compute (Outswapped)             1 |*
Current Process                  1 |*
                                 + - - - - + - - - - + - - - - + - - - - -+
```
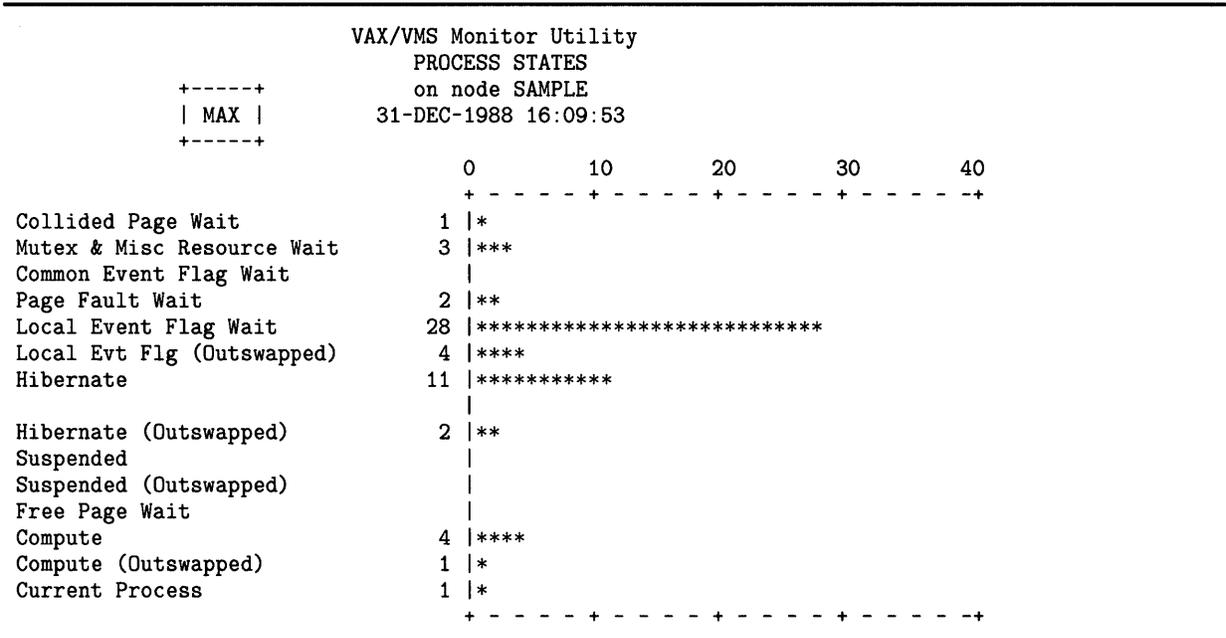
3.3.2 Multiple-Statistic Screen for System Classes

This tabular-style screen is used whenever all four statistics are requested with the /ALL class-name qualifier. Example MON–2 shows a multiple-statistic screen. The precision of the data items is seven whole and two decimal places. For each class, the screen format remains the same with different heading and data item descriptions.

If you request the display of percentages, as in Example MON–3, the percent sign (%) appears in the title and the headings, and the figures consist of three whole and one decimal place.

Example MON–2   Sample Multiple-Statistic Screen

```
                        VAX/VMS Monitor Utility
                       PAGE MANAGEMENT STATISTICS
                            on node SAMPLE
                         31-DEC-1988 16:13:38
                            CUR       AVE       MIN       MAX

    Page Fault Rate         58.00     38.33     18.66     58.00
    Page Read Rate          18.00     16.33     14.66     18.00
    Page Read I/O Rate       3.33      3.16      3.00      3.33
    Page Write Rate         45.00     22.50      0.00     45.00

    Page Write I/O Rate      1.66      0.83      0.00      1.66
    Free List Fault Rate    26.33     15.66      5.00     26.33
    Modified List Fault Rate 4.66      3.83      3.00      4.66
    Demand Zero Fault Rate  12.00      7.66      3.33     12.00
    Global Valid Fault Rate 11.33      7.83      4.33     11.33

    Wrt In Progress Fault Rate 0.00    0.00      0.00      0.00
    System Fault Rate       24.33     12.83      1.33     24.33
    Free List Size        3356.00   3321.50   3287.00   3356.00
    Modified List Size       1.00     70.00      1.00    139.00
```

Example MON–3   Sample Multiple-Statistic Screen (Data Expressed as Percentages)

```
                        VAX/VMS Monitor Utility
                      TIME IN PROCESSOR MODES (%)
                            on node SAMPLE
                         31-DEC-1988 16:13:38
                            CUR%      AVE%      MIN%      MAX%

    Interrupt Stack         20.3      21.9      20.3      23.6

    MP Synchronization       0.0       0.0       0.0       0.0

    Kernel Mode             23.0      23.8      23.0      24.6

    Executive Mode           3.0       3.5       3.0      24.6

    Supervisor Mode          0.0       0.0       0.0       0.6

    User Mode               51.3      46.9      42.6      51.3

    Compatibility Mode       2.3       3.6       0.0       3.9

    Idle Time                0.0       0.0       0.0      94.9
```

3.3.3   Component Classes Screen

For all component classes except RMS and MODES, only one data item for each component is displayed on each screen. The item is identified in the upper left of the screen. Components for which statistics are reported appear in the left column of the screens. If more than one item keyword is specified with the /ITEM qualifier or if /ITEM=ALL is specified, a new screen appears for each item selected. For example, the following command would produce the output of the format shown in Example MON–4:

MONITOR DISK/ITEM=(OPERATION_RATE,QUEUE_LENGTH)

MONITOR Description

Example MON–4 Sample Component Statistics Screens

```
                        VAX/VMS Monitor Utility
                          DISK I/O STATISTICS
                            on node SAMPLE
                         31-DEC-1988 20:08:42

I/O Operation Rate                   CUR       AVE       MIN       MAX

DRA2:          SAMPLEPAGE            0.00      0.03      0.00      0.33
DRB1:          ACCREG               0.00      0.00      0.00      0.00
DRC3:          VMS_X2OR             1.99      0.19      0.00      1.99
DRC4:          SAMPLESECD01         0.00      0.00      0.00      0.00
DBA3:          UMASTER              0.00      0.00      0.00      0.00
DBA5:          MIDNITE              0.00      0.00      0.00      0.00
DRA7:          RES26APR             0.00      0.00      0.00      0.00
DUA4:          RES06AUG             0.00      0.00      0.00      0.00
DUA5:          VMSDOCLIB            0.00      0.00      0.00      0.00
DUA7:          OLD_QVSS$            0.00      0.00      0.00      0.00


                        VAX/VMS Monitor Utility
                          DISK I/O STATISTICS
                            on node SAMPLE
                         31-DEC-1988 20:08:45

I/O Request Queue Length             CUR       AVE       MIN       MAX

DRA2:          SAMPLEPAGE            0.00      0.00      0.00      0.00
DRB1:          ACCREG               2.00      1.43      0.00      4.00
DRC3:          VMS_X2OR             0.00      0.00      0.00      0.00
DRC4:          SAMPLESECD01         0.00      0.00      0.00      0.00
DBA3:          UMASTER              0.00      0.00      0.00      0.00
DBA5:          MIDNITE              0.00      0.00      0.00      0.00
DRA7:          RES26APR             0.00      0.00      0.00      0.00
DUA4:          RES06AUG             0.00      0.00      0.00      0.00
DUA5:          VMSDOCLIB            0.00      0.00      0.00      0.00
DUA7:          OLD_QVSS$            0.00      0.00      0.00      0.00
```

3.4 Recording File Output

A recording file is a VAX RMS sequential disk file that is created when a MONITOR request includes the /RECORD qualifier. A record of binary performance data is written to this file once per interval for each requested class; the record contains a predefined set of data for each of the requested performance classes. The file is created when a MONITOR request is initiated and closed when the request terminates. The resulting file can be used as a source file by later requests to format and display the data on a terminal, to create a summary file, or to record a new recording file with different characteristics.

All data pertaining to the class is recorded, even if you are concurrently displaying only a single statistic or a single item of a component statistics class.

3.4.1 Disk Space for Recording Files

When recording is active (or display output is being routed to a disk file), you can use large quantities of disk space in a short period of time. In particular, if disk quota is exceeded during execution of a MONITOR request, open files are closed, and the request is terminated prematurely. To avoid this situation, use the information provided in Appendix A to estimate the amount of disk space required.

When SYSTEM class data is recorded, the MODES, STATES, and PROCESSES classes are also recorded, even if not specifically requested. When CLUSTER class data is recorded, the MODES and DISK classes are also recorded. To estimate disk space requirements for CLUSTER recording files, multiply the totals for these classes by the number of nodes being monitored. After estimating disk space requirements, check the amount of disk quota available, and set appropriate values for /INTERVAL and /ENDING.

Refer to Appendix A for exact recording file record sizes.

3.4.2 Recording File Version Compatibility

Before Version 5.0 MONITOR can read recording files generated by previous MONITOR versions, you must convert the files to the current format. Use the CONVERT command described in the Commands section of this document. If you specify a list of recording files to produce a multifile summary, all recording files must have the same format.

3.5 Summary Output

Summary output is an ASCII disk file consisting of one display screen image for each requested class. The screen format for each class is based on the statistic requested. The only difference in format between a display screen and a summary screen image is that the word SUMMARY appears in the heading along with a beginning and ending time for the period covered by the summary. The data contained in the summaries is identical to that shown on the final display screen (if display output was also requested) for all except the PROCESSES/TOP, SYSTEM, and CLUSTER summaries.

Since the summary file reflects the accumulation of data throughout the MONITOR request, the average, minimum, and maximum statistics are of particular interest. For the TOP summaries of the PROCESSES and SYSTEM classes, the data represents the top users for the entire duration of the MONITOR request, subject to the following restriction. To be eligible for inclusion in the list of top users, a process must be present and swapped in at the beginning and end of the MONITOR request.

3.6 Multifile Summary Reports

Multifile summary reports provide a convenient method of combining data from a number of recording files to compare average performance statistics (excluding the PROCESSES and CLUSTER classes) for discrete time segments. Use the /BEGINNING and /ENDING command qualifiers to delimit the desired time segment (see the Qualifiers section).

MONITOR Description

To request a multifile summary, use the /SUMMARY command qualifier, and specify a list of recording files with the /INPUT qualifier. Note that since only AVERAGE statistics are collected, you should not specify class-name qualifiers. Note also that multifile summaries are static; that is, they do not provide continuously updated displays.

Caution: Version 5.0 MONITOR file structure must be common to all recording files in the list.

3.6.1 Interpreting Multifile Summary Reports

Multifile summary reports differ from regular (single-file) reports in both content (only AVERAGE statistics are collected) and format. MONITOR formats multifile report data as follows:

- By file—This is the default format. For each class requested, the report displays one column of AVERAGE statistics per input file, along with beginning and ending times for each file. For files that contain data for multiple nodes, there is one column per node per file.

- By node—To request this format, specify the /BY_NODE command qualifier (with the /SUMMARY and /INPUT qualifiers) when you create the summary file. The report combines data for a given node from all files into a single column that shows the average statistic for each data item. The contribution of the data from each file is weighted by the amount of time over which the data was collected (for rate items) or by the number of collections (for level items).

For both formats, MONITOR provides Row Sum, Row Average, Row Maximum, and Row Minimum statistics. These represent simple arithmetic operations performed on all averages in each row of the report.

Note: Because multifile summary reports frequently contain large amounts of data and use a 132-character format, you may want to print them. A single page can display only five columns of data. Depending on the number of recording files, nodes, and classes specified, a report may extend over many pages. In that event, Row values appear on the final page.

The following examples illustrate differences between single-file and multifile reports. In the first example, two fragments of single-file reports are generated on the same node from two different recording files, which cover, respectively, a two-hour and a ten-hour period:

```
                VAX/VMS Monitor Utility
                PAGE MANAGEMENT STATISTICS
                      on node BLUE          From:  31-DEC-1988 09:00:00
                         SUMMARY            To:    31-DEC-1988 11:00:00

                         CUR         AVE         MIN         MAX
Page Fault Rate          ----       90.00        ----        ----


                VAX/VMS Monitor Utility
                PAGE MANAGEMENT STATISTICS
                      on node BLUE          From:  31-DEC-1988 11:00:00
                         SUMMARY            To:    31-DEC-1988 21:00:00

                         CUR         AVE         MIN         MAX
Page Fault Rate          ----        6.00        ----        ----
```

In the next example, the corresponding "by node" multifile report fragment for the entire period shows that the average Page Fault Rate is weighted toward the figure that represents the larger elapsed time:

```
+-----+              VAX/VMS Monitor Utility
| AVE |              PAGE MANAGEMENT STATISTICS
+-----+                  Multifile SUMMARY


           Node:       BLUE (2)
           From: 31-DEC-1988 09:00      Row      Row       Row       Row
           To:   31-DEC-1988 21:00      Sum    Average   Minimum   Maximum

Page Fault Rate               20.00    20.0     20.0     20.00     20.00
```

The Row statistics provided in multifile reports are not time-weighted; they are meaningful only when all input files that contribute to the report cover a common time period, as in Example MON–6 ("by node" report for nodes MOE, CURLEY, and LARRY) in Section 3.6.2. Thus, as the following example shows, Row statistics would not be useful in a "by file" report for node BLUE because the contributing files covered *different* time periods:

```
+-----+              VAX/VMS Monitor Utility
| AVE |              PAGE MANAGEMENT STATISTICS
+-----+                  Multifile SUMMARY

     Node:       BLUE                 BLUE
     From: 31-DEC-1988 09:00   31-DEC-1988 11:00    Row     Row       Row       Row
     To:   31-DEC-1988 11:00   31-DEC-1988 21:00    Sum   Average   Minimum   Maximum

Page Fault Rate       90.00                 6.00   96.0    48.0      6.00     90.00
```

The preceding examples illustrate how MONITOR formats multifile report data. Each column of averages is headed by the name of the node on which the data was collected, the requested beginning and ending times, and (for reports by node) a parenthesized number signifying the number of input files that contributed data to the column. If no explicit beginning and ending times are specified in the summary request, the times stored in the files are listed.

Note that if a column in a report by node contains data from more than one input file, the earliest beginning time and the latest ending time among all the time stamps in those files are listed. Therefore, exercise caution in interpreting "by node" reports because they may include (in the listed time) the range of a period during which no data was collected.

3.6.2 Using Multifile Summary Reports in Single-Node Environments

In single-node environments, reports by file and reports by node have the following applications:

- By file—You can use this type of report (which contains one column of average statistics per input file) when you want to compare data from several different time segments. The input files selected must already contain the time segments of interest when the multifile summary is run; the /BEGINNING and /ENDING qualifiers cannot be used for this purpose because they can define only one time segment. If necessary, execute a preliminary MONITOR command to rerecord an input file, and write data for the desired time segment to a new file.

  Example MON–5 shows a multifile report generated from three input files, each of which contained data for the period 8 A.M. to noon on a different day. Some of these files may have been created as a result of rerecording larger files to extract only the 8 A.M. to noon time segment.

MONITOR Description

Note that the average page fault rate for a morning during the week of 11 May was 28.4, and that the daily averages for that time period ranged from 21.74 to 39.53.

- By node—If you specify the /BY_NODE qualifier, you can combine data from several input files to produce a single average statistic for each data item. If continuous recording is implemented, so that MONITOR data is recorded whenever the system is up, you can produce a report that shows averages for any time segment, regardless of how many input files are included (up to the maximum of 5,000). Use the /BEGINNING and /ENDING qualifiers to define the time segment of interest.

Example MON-5 shows a typical multifile summary report by file for three input files on a single node. The summary was requested with the following command:

MONITOR /INPUT=(11MAY,12MAY,13MAY)/NODISPLAY/SUMMARY PAGE

Example MON–5 Sample Single-Node Multifile Summary

```
+-----+        VAX/VMS Monitor Utility
| AVE |        PAGE MANAGEMENT STATISTICS
+-----+           Multifile SUMMARY
```

	Node: YELLOW	YELLOW	YELLOW				
	From: 11-MAY-1988 08:00	12-MAY-1988 08:00	13-MAY-1988 08:00	Row	Row	Row	Row
	To: 11-MAY-1988 12:00	12-MAY-1988 12:00	13-MAY-1988 12:00	Sum	Average	Minimum	Maximum
Page Fault Rate	39.53	23.98	21.74	85.2	28.4	21.74	39.53
Page Read Rate	14.12	3.30	2.21	19.6	6.5	2.21	14.12
Page Read I/O Rate	1.73	0.52	0.43	2.7	0.9	0.43	1.73
Page Write Rate	11.21	1.54	1.11	13.7	4.6	1.11	11.21
Page Write I/O Rate	0.11	0.01	0.01	0.1	0.0	0.00	0.11
Free List Fault Rate	10.28	5.00	4.73	20.0	6.6	4.73	10.28
Modified List Fault Rate	5.51	4.77	4.53	14.8	4.9	4.53	5.51
Demand Zero Fault Rate	11.58	5.06	4.21	20.8	6.9	4.21	11.58
Global Valid Fault Rate	10.25	8.54	7.76	26.5	8.8	7.76	10.25
Wrt In Progress Fault Rate	0.03	0.01	0.01	0.0	0.0	0.01	0.03
System Fault Rate	0.04	0.09	0.05	0.2	0.0	0.04	0.09
Free List Size	6829.60	7102.33	7271.85	21203.7	7067.9	6829.60	7271.85
Modified List Size	257.31	210.41	216.91	684.6	228.2	210.41	257.31

3.6.3 Using Multifile Summary Reports in Cluster Environments

In cluster or other multinode environments (DECnet or any set of noncommunicating nodes), reports by file and by node have the following applications:

- By file—You can use this type of report to compare several time segments as you would for the single-node case. By selecting the appropriate input files, you can include in the report data from any single node or set of nodes.

- By node—As in the single-node case, you can use this type of report to combine data from several input files for a given node in one column of averages. Note, however, that there is one column for each node. By specifying the /BEGINNING and /ENDING qualifiers to select a common time period, you can use the report to make node-to-node comparisons. In addition, you can examine the "Row" statistics to evaluate overall cluster performance.

For cluster environments only, you can use the /NODE qualifier to select data for one or more nodes from the set of input files. You can combine data for those nodes by specifying the /BY_NODE qualifier.

In Example MON-6, consider the Idle Time item from the MODES class. The Row Sum figure (116.2) represents the average idle time for the entire cluster over the two-hour period covered by the report; the Row Average (38.7) is the average idle time per node over the same time period. The Row Minimum (0.02) and Row Maximum (92.63) figures identify the lowest and highest of the individual node averages.

Example MON-6 shows a typical cluster multifile summary. It was generated by the following command:

```
MONITOR/INPUT=(MOE.DAT;*,CURLEY.DAT;*,LARRY.DAT;*) MODES,PAGE -
/SUMMARY/BY_NODE /NODISPLAY/BEGINNING="18:17"/ENDING="20:17"
```

Example MON-6 Sample Cluster Multifile Summary

```
      +-----+       VAX/VMS Monitor Utility
      | AVE |       TIME IN PROCESSOR MODES
      +-----+        MULTI-FILE SUMMARY
      Node:    MOE             CURLEY (2)       LARRY
      From: 31-DEC-1988 18:17  31-DEC-1988 18:17  31-DEC-1988 18:17   Row     Row      Row      Row
      To:   31-DEC-1988 20:17  31-DEC-1988 20:17  31-DEC-1988 20:17   Sum   Average  Minimum  Maximum
Interrupt Stack        6.51             0.50             6.25      13.2     4.4     0.50      6.51

MP Synchronization     0.00             0.00             0.00       0.0     0.0     0.00      0.00
Kernel Mode           25.73             0.42            12.43      38.5    12.8     0.42     25.73

Executive Mode         9.46             0.95             1.81      12.2     4.0     0.95      9.46

Supervisor Mode        1.97             0.00             0.16       2.1     0.7     0.00      1.97

User Mode             32.72             5.33            79.32     117.3    39.1     5.33     79.32

Compatibility Mode     0.00             0.07             0.00       0.0     0.0     0.00      0.07

Idle Time             23.61            92.63             0.02     116.2    38.7     0.02     92.63
      +-----+       VAX/VMS Monitor Utility
      | AVE |       PAGE MANAGEMENT STATISTICS
      +-----+        MULTI-FILE SUMMARY

      Node:    MOE             CURLEY(2)        LARRY
      From: 31-DEC-1988 18:17  31-DEC-1988 18:17  31-DEC-1988 18:17   Row     Row      Row      Row
      To:   31-DEC-1988 20:17  31-DEC-1988 20:17  31-DEC-1988 20:17   Sum   Average  Minimum  Maximum

Page Fault Rate           36.73             8.81             0.49     46.0    15.3     0.49     36.73
Page Read Rate            14.28             4.71             0.00     19.0     6.3     0.00     14.28
Page Read I/O Rate         1.20             0.70             0.00      1.9     0.6     0.00      1.20
Page Write Rate            0.00             0.00             0.00      0.0     0.0     0.00      0.00
Page Write I/O Rate        0.00             0.00             0.00      0.0     0.0     0.00      0.00

Free List Fault Rate       8.60             1.40             0.24     10.2     3.4     0.24      8.60
Modified List Fault Rate   5.83             2.29             0.00      8.1     2.7     0.00      5.83
Demand Zero Fault Rate    12.96             1.68             0.24     14.8     4.9     0.24     12.96
Global Valid Fault Rate    8.10             2.69             0.00     10.8     3.6     0.00      8.10
Wrt In Progress Fault Rate 0.00             0.00             0.00      0.0     0.0     0.00      0.00
System Fault Rate          4.92             0.53             0.18      5.6     1.8     0.18      4.92

Free List Size          7586.30          8630.14          9665.06  25881.5  8627.1  7586.30  9665.06
Modified List Size        87.69           324.07            32.12    443.8   147.9    32.12   324.07
```

Sample command procedures for generating multifile summaries in a cluster environment are provided in the Examples section of this document.

4	**Error Messages**

The *VMS System Messages and Recovery Procedures Reference Volume* lists the messages generated by MONITOR and provides explanations and suggested user actions.

MONITOR Usage Summary

The Monitor Utility (MONITOR) is a system management tool that enables you to obtain information on operating system performance.

FORMAT	**MONITOR** *[classname [,...]]*

COMMAND PARAMETER

classname[,...]

Specifies the class of performance data to be monitored. To monitor all classes, specify the ALL_CLASSES parameter. When you specify several specific classes, separate the class-name parameters with commas or plus signs. You cannot specify the CLUSTER class name with any other class name. Cluster monitoring functions require that DECnet-VAX be installed.

You must specify one or more of the following parameters:

ALL_CLASSES	Statistics for all classes
CLUSTER	Clusterwide performance statistics
DECNET	DECnet-VAX statistics
DISK	Disk I/O statistics
DLOCK	Distributed lock management statistics
FCP	File control primitive statistics
FILE_SYSTEM_CACHE	File system cache statistics
IO	System I/O statistics
LOCK	Lock management statistics
MODES	Time spent in each of the processor modes
MSCP_SERVER	MSCP server statistics
PAGE	Page management statistics
POOL	Statistics on space allocation in the nonpaged dynamic pool
PROCESSES	Statistics on all processes
RMS	Record Management Services statistics
SCS	System communications services statistics
STATES	Number of processes in each of the scheduler states
SYSTEM	Summary of statistics from other classes

MONITOR class names accept a variety of qualifiers, as summarized in Table MON-1 in the Description section. Complete descriptions of qualifiers for each class are provided in the Commands section.

usage summary

To invoke MONITOR, enter the DCL command MONITOR. The utility then displays the following prompt:

MONITOR>

MONITOR Usage Summary

In response to the prompt, you can enter any utility command.

You can also initiate MONITOR requests from command level by entering the DCL command MONITOR with the desired qualifiers and parameters. However, in terms of system overhead, it is preferable to initiate requests in response to the MONITOR> prompt.

Generally, each MONITOR request runs until the time specified or implied by the /ENDING qualifier. However, you can press CTRL/C or CTRL/Z to terminate a MONITOR request earlier. Pressing CTRL/C terminates the current request without exiting from the utility. You can then initiate a new request or enter any MONITOR command. Pressing CTRL/Z terminates the current request and exits from MONITOR.

Information collected by MONITOR is normally displayed as ASCII screen images. You can use the optional /DISPLAY qualifier to specify a disk file to contain the information. If you omit the file specification, output is directed to SYS$OUTPUT. See the Qualifiers section for a discussion of the /DISPLAY qualifier.

MONITOR QUALIFIERS

This section describes qualifiers for the MONITOR and SET DEFAULT commands. Note that these commands accept the same qualifiers. The qualifiers follow the standard rules of DCL grammar, as specified in the *VMS DCL Concepts Manual*. Thus, you can abbreviate any qualifier or keyword as long as the abbreviation is not ambiguous. Use the asterisk and the percent sign as wildcard characters unless otherwise noted.

/BEGINNING

Specifies the time that monitoring begins.

FORMAT   **/BEGINNING=***time*

PARAMETERS   *time*
Specifies a combination of absolute and delta times. Observe the syntax rules for time values described in the *VMS DCL Concepts Manual*.

DESCRIPTION   If you are monitoring a running system, and you omit the /BEGINNING qualifier, monitoring begins when you enter the MONITOR command. However, if you have also specified the /INPUT qualifier to play back data from an input recording file, /BEGINNING defaults to the beginning time recorded in the input file. If you specify /BEGINNING with a time, but are playing back a recording file, MONITOR selects the later of the beginning time of the file and the beginning time you specify. If you are monitoring a remote node, the local node time is used to determine beginning time.

If you specify a future time for a request to monitor a running system, MONITOR issues an informational message, and the process issuing the request hibernates until the specified time. This feature can be useful when you run MONITOR from a batch job.

/BY_NODE

Specifies that performance class data in a multifile summary be displayed as a single column of AVERAGE statistics for each node.

FORMAT   **/[NO]BY_NODE**

DESCRIPTION   The /BY_NODE qualifier can be specified only in combination with the /SUMMARY qualifier. One column of AVERAGE statistics per node appears for each class requested.

By default, multifile summaries include one column of AVERAGE statistics for each node requested in each input file.

/COMMENT

Specifies an ASCII string to be stored in the output recording file. The string can contain up to 60 characters.

FORMAT **/COMMENT=**_"string"_
/NOCOMMENT

PARAMETERS _"string"_
A comment string up to 60 characters in length.

DESCRIPTION The /COMMENT qualifier is valid only when /RECORD is also specified. If you omit the qualifier or specify /NOCOMMENT, a string consisting of 60 blanks is stored in the recording file by default. When a recording file containing a comment is played back, the comment is included in the heading of the display or single-file summary.

/DISPLAY

Specifies whether information collected by MONITOR is to be displayed as ASCII screen images. Optionally names the disk file to contain the output.

FORMAT   **/DISPLAY**[*=file-spec*]
/NODISPLAY

PARAMETERS   *[=file-spec]*
The specification of the file to which output is directed.

DESCRIPTION   If you omit the optional file specification, output is written to SYS$OUTPUT.

By default, display output is produced. Note, however, that display output is never produced when a multifile summary is requested. Sections 3.1 to 3.3 describe display output formats.

/ENDING

Specifies the time that monitoring ends.

FORMAT   /ENDING=*time*

PARAMETERS   *time*
Specifies a combination of absolute and delta times. Observe the syntax rules for time values described in the *VMS DCL Concepts Manual*.

DESCRIPTION   If you are monitoring a running system and omit the /ENDING qualifier, monitoring continues until you terminate the request with CTRL/C or CTRL/Z. If you have also specified the /INPUT qualifier to play back data from an input recording file, /ENDING defaults to the ending time recorded in the input file. If you specify /ENDING with a time, but are playing back a recording file, MONITOR selects the earlier of the ending time of the file and the ending time you specify. For live requests, the local node's time stamp is used to determine ending time.

You can prematurely terminate a request, regardless of the value of the /ENDING qualifier, by pressing CTRL/C or CTRL/Z. To prematurely terminate a request running in a noninteractive process (that is, a batch job or a detached process or subprocess), enter the appropriate DCL command to terminate the process.

/FLUSH_INTERVAL

Specifies the interval, in seconds, at which data collected by MONITOR (contents of MONITOR buffers) is written to disk. Values must be in the range of 1 through 9,999,999. The default interval is 300 seconds.

FORMAT   **/FLUSH_INTERVAL=***seconds*

PARAMETERS   *seconds*
A value in the range of 1 through 9,999,999.

DESCRIPTION   If you are writing data to a shared recording file currently in use, you should specify a short interval to ensure that others accessing the file receive data that is as current as possible. The smaller the interval, the less data is lost if a system failure occurs while recording.

/INPUT

Controls whether performance data is played back from one or more input files or collected from the running system.

FORMAT **/INPUT***[=(file-spec,...)]*
/NOINPUT

PARAMETERS ***file-spec(,...)***
Specifies one or more input files. If you specify more than one file, enclose the list in parentheses, and separate the file specifications with commas. Wildcard characters are allowed in the file specification.

Caution: **Version 5.0 MONITOR file structure must be common to all recording files in the list.**

If you want to include files generated by previous MONITOR versions, you must first convert the files to the current format using the CONVERT command described in the Commands section.

DESCRIPTION With multiple input files, you must use the /SUMMARY qualifier. The maximum number of files MONITOR accepts for a multifile summary is 5,000.

In a list of input files, any omitted segment of the file specification (name or type) is defaulted to the corresponding segment of the previous file specification.

If you omit the file type, and you have not specified the file type previously in an input file list, the default file type DAT is used. If you omit the file specification, MONITOR assigns the default file name MONITOR.DAT. The current device and directory defaults are applied.

If you omit the qualifier, performance data is collected from the running system.

/INTERVAL

Specifies the sampling interval between data collection events, recording events, and display events.

FORMAT	**/INTERVAL=**_seconds_

PARAMETERS

seconds

A value in the range of 1 through 9,999,999.

DESCRIPTION

Collection events, recording events, and display events occur within a MONITOR request. Use the /INTERVAL qualifier to control the frequency of these events. A collection event causes raw data for all requested classes to be collected from the operating system or from a previously recorded file. A recording event causes data for all requested classes to be written to a recording file. A display event causes a screen image to be composed, for a single class, from the accumulated data collected for that class since the beginning of the MONITOR request.

For live collection requests, a collection event is always followed immediately by a recording event (if requested). The frequency of collection/recording event pairs is controlled by the /INTERVAL qualifier, which specifies the number of seconds that must elapse between occurrences of the event pair. Display events occur asynchronously to collection/recording event pairs at a frequency governed by the /VIEWING_TIME qualifier.

For playback requests, a collection event occurs each time a new interval is encountered in the input file of previously recorded data. A recording event (if requested) does not necessarily follow immediately as it does in live collection. Its frequency is still governed by the /INTERVAL qualifier; the specified /INTERVAL value is interpreted in terms of the /INTERVAL value specified when the input file was created. The new value must be an integral multiple of the original value. A recording event is then triggered every time an interval is encountered in the input file that is the appropriate multiple of the original interval. For playback requests, occurrences of display events (if requested) are indicated in exactly the same way as recording events (with the /INTERVAL qualifier) and immediately follow recording events (if both are specified). The actual length of time a displayed image remains on the screen is still specified with the /VIEWING_TIME qualifier, but, unlike the live collection case, this qualifier is not used to signal a display event. Table MON-2 summarizes which qualifiers cause the various MONITOR events.

Table MON–2 Relationship of MONITOR Command Qualifier to Event

Event	Live Collection Qualifier	Playback Qualifier
Collection	/INTERVAL	Original /INTERVAL value (from file)
Recording	/INTERVAL	/INTERVAL
Display	/VIEWING_TIME	/INTERVAL

Note that, for live requests, the collection interval is defined as the number of seconds from the end of one collection event to the beginning of the next. A collection event includes collection for all requested classes on all nodes specified. (For multiple-node requests, a collection event must complete on all nodes before a new event is initiated.) Therefore, the elapsed time from the beginning of one collection event to the beginning of the next is the interval value plus the time it takes to do the collection. For some requests, notably those including many classes or the PROCESSES, RMS, CLUSTER, or SYSTEM classes, collection time can be significant.

For /INPUT requests, the interval value defaults to the value specified in the input recording file. The default for monitoring the running system is three seconds for all classes except ALL_CLASSES, CLUSTER, and SYSTEM, which have a default of six seconds.

/NODE

Specifies the nodes (up to 48 in a cluster) for which data is to be collected.

FORMAT **/NODE**=*(nodename,...)*

PARAMETERS *nodename(,...)*
Specifies one or more node names. If you specify more than one name, separate the names with commas, and enclose the list in parentheses.

DESCRIPTION You can specify a maximum of 48 node names with the /NODE qualifier.

If you specify multiple node names with multiple system classes, MONITOR displays one class at a time for each node. For example, the command MONITOR/NODE=(NODE_A,NODE_B) STATES,MODES generates STATES data for NODE_A and NODE_B and then MODES data.

Note that if you specify the /ITEM class-name qualifier with the DISK or SCS classes, MONITOR displays one item at a time for each node.

/OUTPUT

Used with the CONVERT command, this qualifier specifies the name of the converted recording file.

FORMAT

/OUTPUT=*file-spec*

PARAMETERS

file-spec
The file specification of the output file. The default specification is MONITOR.DAT. File lists are not permitted.

DESCRIPTION

Recording files produced using earlier MONITOR versions must be converted to the current format before they can be played back with Version 5.0 MONITOR.

/RECORD

Specifies that a binary disk file be created containing all collected data for the request.

FORMAT   **/RECORD**_[=file-spec]_
              /NORECORD

PARAMETERS *file-spec*
The name of the recording file. Note that recording is restricted to files on disks. No wildcard characters are allowed in the file specification. If you omit the file type, the default file type is DAT. If you omit the file specification, output is generated to a file named MONITOR.DAT in the current default device and directory. If you specify an existing file but omit the version number, a new version of the file is created.

DESCRIPTION The output consists of all data for the requested classes, regardless of the class-name qualifiers specified. Note that recording file output is not produced when a multifile summary is requested.

By default, no recording file output is produced.

For more information on MONITOR recording file output, see Section 3.4. Recording file record formats are described in Appendix A.

/SUMMARY

Specifies that an ASCII disk file be created containing summary statistics on all data collected for this request.

FORMAT   **/SUMMARY**[*=file-spec*]
               /NOSUMMARY

PARAMETERS *file-spec*
The name of the summary file. If the optional file specification is omitted, it defaults to MONITOR.SUM. By default, no summary output is produced.

DESCRIPTION The summary file, generated at the end of monitoring, contains one or more pages of output for each requested class. The format of each page is similar to that of display output and is determined by the class-name qualifiers. The /ALL qualifier is applied to all class names for which no other qualifier is specified.

Summary output formats are described in Section 3.5; multifile summaries are described in Section 3.6.

/VIEWING_TIME

For /DISPLAY requests, specifies the duration for each screen image display.

FORMAT   **/VIEWING_TIME=**seconds

PARAMETERS   **seconds**
A value in the range of 1 through 9,999,999.

DESCRIPTION   If you are monitoring the running system, /VIEWING_TIME defaults to the /INTERVAL value. If you specify /INPUT, and you are monitoring a recording file, /VIEWING_TIME defaults to three seconds.

Effective viewing time varies, however, depending on whether you are running MONITOR on your local system or on a remote node. (*Remote* in this context refers to the use of the SET HOST command to access another node.) For remote access, the time required to display the screen is included in the viewing time, while for local access, this time is not included. Therefore, use a larger viewing time than the 3-second default when running MONITOR on a remote system. The value appropriate for remote access depends on your terminal baud rate. For a 9600–baud terminal line, six seconds is a reasonable viewing time.

Note also that the time between full screens of data for the PROCESSES display is controlled by this qualifier.

MONITOR COMMANDS

This section describes MONITOR commands. For commands that specify class-name parameters (other than ALL—CLASSES), a sample display or summary of each class is provided, with a brief description of the items in the class.

The commands follow the standard rules of DCL grammar, as specified in the *VMS DCL Concepts Manual*. Thus, you can abbreviate any command as long as the abbreviation is not ambiguous.

MONITOR recognizes the exclamation point (!) as a comment character. Thus, full- or partial-line comments are acceptable in command files specified as input to MONITOR.

CONVERT

The CONVERT command converts a pre-Version 5.0 MONITOR recording file to the current format.

FORMAT   **CONVERT** *file-spec*

PARAMETERS   *file-spec*
Specifies the file to be converted. The default file specification is MONITOR.DAT.

QUALIFIERS   */OUTPUT*
The file specification of the converted file. The default specification is MONITOR.DAT.

DESCRIPTION   You must convert pre-Version 5.0 recording files to the current format before attemping to play them back with Version 5.0 MONITOR.

EXAMPLE

```
MONITOR> CONVERT 24MAY_MONITOR.DAT/OUTPUT=24MAY_NEWMON.DAT
```

The command in this example converts the file 24MAY_MONITOR.DAT to the current format and names the output file 24MAY_NEWMON.DAT.

EXECUTE (@)

The EXECUTE (@) command executes a series of MONITOR commands contained in a file.

FORMAT   **EXECUTE (@)** *file-spec*

PARAMETERS *file-spec*
Specifies a command file to be executed by the EXECUTE (@) command.

QUALIFIERS *None.*

DESCRIPTION With the EXECUTE command, you can direct MONITOR to obtain command input from a specified file rather than from the terminal. The file can contain any valid MONITOR command except an EXECUTE (@) command. Commands in the file are executed sequentially. If you omit the optional file specification, the default is MONITOR.MON.

After the file has executed, commands are obtained from the terminal.

EXAMPLE

```
MONITOR> EXECUTE INQMEM.MON
   .
   .
   .
MONITOR> MONITOR /RECORD
```

Contents of the file INQMEM.MON are as follows:

```
! This file sets defaults for a memory management inquiry using
! INTERVAL=5, PAGE, IO, and PROCESSES/TOPFAULT
!
       .
       .
       .
SET DEFAULT /INTERVAL=5 PAGE, IO, PROCESSES/TOPFAULT
```

In this example, appropriate default values for a memory management investigation are established in the file INQMEM.MON, and the file is executed with the EXECUTE command. Then a subsequent MONITOR command uses those defaults, adding the /RECORD qualifier, to display and record the selected classes with a 5-second interval.

Note that the defaults established when the file INQMEM.MON is executed remain in effect until changed explicitly or until you exit from the utility.

EXIT

The EXIT command terminates MONITOR, returning control to command level.

FORMAT   **EXIT**

PARAMETERS   *None.*

QUALIFIERS   *None.*

HELP

The HELP command displays information about MONITOR.

FORMAT

HELP *[command]*

PARAMETERS

command
Specifies the name of a MONITOR command for which HELP is desired.

QUALIFIERS

None.

EXAMPLE

```
MONITOR> HELP INITIALIZE

INITIALIZE

  Re-establish initial default values for MONITOR qualifiers and class-name
  parameters.

Topic?
```

The command in this example requests help information about the INITIALIZE command.

INITIALIZE

The INITIALIZE command reestablishes initial default settings for parameters and qualifiers previously altered by the SET DEFAULT command.

FORMAT	**INITIALIZE**

PARAMETERS	*None.*

QUALIFIERS	*None.*

MONITOR ALL_CLASSES

The MONITOR ALL_CLASSES command initiates monitoring of statistics for all classes except the CLUSTER and RMS classes.

FORMAT	**MONITOR ALL_CLASSES**

QUALIFIERS	*/qualifier[,...]* One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL

Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE

Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT

Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM

Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM

Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION

If you do not specify any qualifiers with the ALL_CLASSES parameter, normal default output is produced for each class. (See Table MON-1.) The qualifiers have no effect on display of the PROCESSES class.

Note that the default interval is six seconds.

The MONITOR ALL_CLASSES command is particularly useful for playback of recording files, since it eliminates the need to specify the particular classes of performance data the recording file contains. To override any of the default qualifiers, specify the class name with the qualifier after specifying ALL_CLASSES.

EXAMPLE

```
MONITOR> MONITOR/INPUT=SYS$MANAGER:LOADBAL.DAT ALL_CLASSES,PROCESSES/TOPCPU
```

This command initiates playback of the recording file
SYS$MANAGER:LOADBAL.DAT. All data contained in the file will be
displayed.

MONITOR CLUSTER

The MONITOR CLUSTER command initiates monitoring of the CLUSTER statistics class, which shows clusterwide CPU, memory, disk, and locking activity.

FORMAT	**MONITOR CLUSTER**

QUALIFIERS	*/qualifier[,...]* One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL

Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE

Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT

Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM

Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM

Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION

For the CLUSTER class, MONITOR collects data items for up to 48 nodes in a cluster. Because this class combines the most significant clusterwide performance statistics in a single display, it is particularly useful to cluster managers and other users seeking an overview of cluster activity.

MONITOR does not recognize nodes that enter the cluster while a request is active. MONITOR therefore does not collect data for these nodes.

You cannot specify the CLUSTER class in the same request with any other class.

The CLUSTER class includes the following data items:

- CPU Busy—Percentage of CPU in use; includes activity in all processor modes (except Idle Time) for each node.

- Percent Memory In Use—Memory in use on each node; calculated by dividing the Free List Size by total available memory and subtracting the result from 100%.

- I/O Operation Rate—Total rate of disk I/O operations on each disk by all nodes currently active in the request.

  In cluster configurations, the MSCP Server software makes locally attached and HSC disks available to other nodes. A node uses *remote* access to a disk when it accesses the disk through another VAX node (using the MSCP Server). A node uses *direct* access to a disk when it directly accesses a locally attached or HSC disk.

  An "R" following the device name indicates that the displayed statistics represent I/O operations requested by nodes using remote access.

  If an "R" does not appear after the device name, the displayed statistics represent I/O operations issued by nodes with direct access. These I/O operations might include those issued by the MSCP Server on behalf of remote requests.

- Total ENQ/DEQ Rate—Sum of all local, incoming, and outgoing ENQ's, DEQ's, and conversions.

Two display formats are provided, depending on the class-name qualifier specified:

- A tabular style format for the /ALL qualifier

- A bar graph style format for the /AVERAGE, /CURRENT, /MAXIMUM, and /MINIMUM qualifiers

Note to Cluster Managers on MONITOR_SERVER Process

When users enter the MONITOR CLUSTER command, MONITOR activates the image SYS$SYSTEM:VPM.EXE, which creates a process called MONITOR_SERVER on each active cluster node. (If users specify the /NODE qualifier with the MONITOR CLUSTER command or with any command of the form MONITOR class name, MONITOR creates the process only on the specified nodes.) The server process gathers data from remote nodes for live display or to record on the local node. To ensure accurate and timely data collection, the process is started at priority 15. Because server processes consume minimal resources, they have no significant effect on system performance.

By default, MONITOR_SERVER processes are started in the system DECNET account, which is created when the NETCONFIG.COM command procedure executes at bootstrap time. If this account is not present on your system, you must either create it by executing NETCONFIG.COM, or specify another account in which the server processes can be started.

If you want to start the processes in another account, use the following sequence of commands to define VPM as known object 51 in the DECnet database and associate the object with the desired account:

```
$ SET PROCESS/PRIVILEGE=SYSPRV
$ RUN SYS$SYSTEM:NCP
NCP> DEFINE OBJECT NUMBER 51 -
_ NAME VPM -
_ FILE SYS$SYSTEM:VPM.EXE -
_ PROXY NONE -
_ ACCOUNT accountname -
_ USERNAME username -
_ PASSWORD password
NCP> SET OBJECT NUMBER 51 -
_ NAME VPM -
_ FILE SYS$SYSTEM:VPM.EXE -
_ PROXY NONE -
_ ACCOUNT accountname -
_ USERNAME username -
_ PASSWORD password
NCP> EXIT
$ SET PROCESS/PRIVILEGE=NOSYSPRV
```

For each server process, MONITOR creates a log file on the local node to which information on server connection activity, including error messages, is written. Note that error messages are written to the file only when errors occur. A single version is maintained for the life of the system. The default file specification has the form SYS$COMMON:[SYSMGR]VPM$nodename.LOG. The node name portion of the specification identifies the node on which the MONITOR_SERVER process has been started.

If you want to change the default specification, you can redefine the executive-mode logical name VPM$LOG_FILE in the system logical name table on the appropriate nodes. For example, if you wanted to write server error logging data to the file WRKD:[MONSERVER]VPM_ERRORS.LOG, you would define VPM$LOG_FILE as follows:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE VPM$LOG_FILE -
_$ WRKD:[MONSERVER]VPM_ERRORS.LOG
```

To direct to a single file data for all MONITOR_SERVER processes on the cluster, you could assign the logical name the same value on each member system. Note that because the log files are created as shared sequential files, multiple server processes can access them simultaneously.

If you routinely monitor your cluster, you can reduce server startup time significantly by creating MONITOR_SERVER processes on each member node at bootstrap time and maintaining the processes for the life of the system. To do so, add the following lines to the appropriate site-independent startup command files:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE VPM$SERVER_LIVE TRUE
$ RUN/DETACH/PAGE_FILE=10000 SYS$SYSTEM:VPM.EXE
```

You can enter these previous commands interactively at any time. Note, however, that you require the following privileges: ALTPRI, NETMBX, PSWAPM, SYSNAM, SYSPRV, and TMPMBX.

EXAMPLE

```
MONITOR> MONITOR CLUSTER/ALL
```

```
                        VAX/VMS Monitor Utility
                          CLUSTER STATISTICS
                           on node CURLEY
                        31-DEC-1988 12:25:13
```

CPU Busy	CUR	AVE	MIN	MAX
LARRY	100.00	100.00	100.00	100.00
CURLEY	100.00	99.83	100.00	100.00
MOE	8.52	8.50	8.52	8.52

```
                        VAX/VMS Monitor Utility
                          CLUSTER STATISTICS
                           on node CURLEY
                        31-DEC-1988 12:25:19
```

%Memory In Use	CUR	AVE	MIN	MAX
MOE	88.00	88.00	88.00	88.00
LARRY	78.00	78.00	77.00	78.00
CURLEY	72.00	72.50	72.00	72.00

```
                        VAX/VMS Monitor Utility
                          CLUSTER STATISTICS
                           on node CURLEY
                        31-DEC-1988 12:25:25
```

I/O Operation Rate			CUR	AVE	MIN	MAX
111DUA7:	(DECEIT)	SQMCLUSTERV4	0.48	6.53	0.48	10.41
111DUA6:	(DECEIT)	QUALD	1.93	1.07	0.00	1.93
111DUA4:	(DECEIT)	PAGESWAPDISK	1.44	0.96	0.00	1.44
111DUA2:	(DECEIT)	TSDPERF	0.32	0.53	0.16	1.12
LARRY$DRA3:		QUALQUEST	0.00	0.21	0.00	0.64
MOE$DMA1:		UVMSQAR	0.00	0.00	0.00	0.00
MOE$DRA5:		USER01	0.00	0.00	0.00	0.00
LARRY$DRA4:		TIMEDEV	0.00	0.00	0.00	0.00
LARRY$DBB3:		REGLIB	0.00	0.00	0.00	0.00
111DUA3:	(DECEIT)	DUMPDISK	0.00	0.00	0.00	0.00
111DUA5:	(DECEIT)	BPMDISK	0.00	0.00	0.00	0.00
111DJA8:	(DECEIT)	ORLEAN	0.00	0.00	0.00	0.00
111DJA10:	(DECEIT)	QMISDATABASE	0.00	0.00	0.00	0.00
111DJA9:	(DECEIT)	MPI$DATA	0.00	0.00	0.00	0.00

```
                        VAX/VMS Monitor Utility
                          CLUSTER STATISTICS
                           on node CURLEY
                        31-DEC-1988 12:25:56
```

Tot ENQ/DEQ Rate	CUR	AVE	MIN	MAX
MOE	7.90	14.92	0.00	43.12
LARRY	20.48	14.64	0.00	46.92
CURLEY	1.93	13.29	0.00	57.30

This example shows the tabular style format for the CLUSTER display.

MONITOR DECNET

The MONITOR DECNET command initiates monitoring of the DECNET class, which includes information on DECnet–VAX network activity.

FORMAT   **MONITOR DECNET**

QUALIFIERS   */qualifier[,...]*
One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL
Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE
Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT
Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM
Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM
Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION   The DECNET class consists of the following data items:

- Arriving Local Packet Rate—Rate at which local packets are being received

- Departing Local Packet Rate—Rate at which local packets are being sent

- Arriving Transit Packet Rate—Rate at which transit packets are arriving

- Transit Congestion Loss Rate—Rate of transit congestion loss

- Receiver Buff Failure Rate—Rate of receiver buffer failures

- LRPs Available—Number of large request packets not in use

EXAMPLE

```
MONITOR> MONITOR DECNET
                         VAX/VMS Monitor Utility
                            DECNET STATISTICS
                             on node SAMPLE
                          31-DEC-1988 22:22:44
                              CUR       AVE       MIN       MAX
     Arriving Local Packet Rate     9.54      5.08      0.00     11.25

     Departng Local Packet Rate     9.22      4.66      0.00     10.92


     Arriving Trans Packet Rate     0.00      0.00      0.00      0.00

     Trans Congestion Loss Rate     0.00      0.00      0.00      0.00


     Receiver Buff Failure Rate     0.00      0.00      0.00      0.00

     LRPs Available                13.00     11.50      9.00     15.00
```

This example shows that arriving and departing network packet rates (including control packets) are roughly equivalent, and that network activity is currently at a level higher than the average since monitoring began, but not at its highest point. Note also that the count of large request packets (LPRs) is displayed; LRPs are used by various components of VAX /VMS, but primarily by DECnet-VAX. A sufficient number of these large request packets must be available to ensure that at peak periods of network use they are not exhausted. If they become depleted, network performance may degrade. The number of packets preallocated at boot time is determined by the SYSGEN parameter LRPCNT.

MONITOR DISK

The MONITOR DISK command initiates monitoring of the DISK statistics class.

FORMAT　　　**MONITOR DISK**

QUALIFIERS　　*/qualifier[,...]*
One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL
Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE
Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT
Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/ITEM=(keyword[,...])
Selects one or more data items for inclusion in display and summary output. If you specify two or more keywords, enclose them in parentheses, and separate them with commas. When the /ITEM qualifier is omitted, the default is /ITEM=OPERATION_RATE.

/MAXIMUM
Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM
Selects minimum statistics to be displayed in a bar graph for display and summary output.

/[NO]PERCENT
Controls whether statistics are expressed as percent values in display and summary output. The /PERCENT qualifier is applicable only to the DISK, MODES, SCS, and STATES classes. By default, statistics are not expressed as percent values.

/ITEM Qualifier Keywords

ALL

Specifies that statistics on all data items collected for the disks are displayed on successive screens.

OPERATION_RATE

Specifies that I/O operation rate statistics are displayed for each disk.

QUEUE_LENGTH

Specifies that the number of I/O request packets being serviced (current or waiting) is displayed for each disk.

DESCRIPTION   The DISK class is a component class. Data items for this class are collected for each mounted disk device in a single-node or cluster system. The DISK class consists of the following data items:

- I/O Operation Rate—Rate at which I/O operations occur on each disk. By comparing operation rates for all disks in the system, you can tell which disks are busy and which are idle. However, since this statistic does not provide information on the time required for individual operations, use discretion in interpreting it.

- I/O Request Queue Length—Number of outstanding I/O request packets. Includes the request currently being serviced and those awaiting service. Note that this item is always sampled at a 1-second interval, regardless of the value specified with the /INTERVAL command qualifier.

In the following example, typical of a cluster environment, note that each disk is identified by three elements:

- Disk name ending in a colon.

- Name of the cluster node through which the disk is accessed. This field appears only in the multiple-statistic display; it is not included in single-statistic displays or multifile summaries.

- Volume label.

In cluster configurations, the MSCP Server software makes locally attached and HSC disks available to other nodes. A node uses *remote* access to a disk when it accesses the disk through another VAX node (using the MSCP Server). A node uses *direct* access to a disk when it directly accesses a locally attached or HSC disk.

An "R" following the device name indicates that the displayed statistics represent I/O operations requested by nodes using remote access.

If an "R" does not appear after the device name, the displayed statistics represent I/O operations issued by nodes with direct access. These I/O operations might include those issued by the MSCP Server on behalf of remote requests.

EXAMPLE

```
MONITOR> MONITOR DISK/ITEM=QUEUE_LENGTH
                         VAX/VMS Monitor Utility
                           DISK I/O STATISTICS
                              on node SAMPLE
                          31-DEC-1988 14:19:56

I/O Request Queue Length                     CUR       AVE       MIN       MAX

SAMPLE$DBA0:              SAMPLE09APR        0.00      0.00      0.00      0.00
SAMPLE$DRA2:              SAMPLEPAGE         2.00      1.43      0.00      4.00
SAMPLE$DRB1:              ACCREG             0.00      0.00      0.00      0.00
$1$DRA5:       (MOE)     MOE$$PAGE          0.00      0.00      0.00      0.00
$1$DBA3:       (CURLEY)  UMASTER            0.00      0.00      0.00      0.00
$1$DBA5:       (CURLEY)  MIDNITE            0.00      0.00      0.00      0.00
$2$DRA7:       (LARRY)   RES26APR           0.00      0.00      0.00      0.00
$2$DRB6:       (LARRY)   CLUSTERDUMP1       0.00      0.00      0.00      0.00
$255$DUA4:     (SHEMP)   RES06AUG           0.00      0.00      0.00      0.00
$255$DUA5:     (SHEMP)   VMSDOCLIB          0.00      0.00      0.00      0.00
```

This example, typical of a cluster environment, shows, for each disk, the number of I/O packets awaiting service or in service. Note that the device SAMPLE$DRA2 is the only device with a nonzero queue length. Since MONITOR samples queue lengths every second, regardless of the collection interval value, the accuracy of the data does not depend on the collection interval.

MONITOR DLOCK

The MONITOR DLOCK command initiates monitoring of the DLOCK (distributed lock management) statistics class.

FORMAT   **MONITOR DLOCK**

QUALIFIERS   */qualifier[,...]*
One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL
Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE
Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT
Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM
Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM
Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION   The DLOCK class is useful for monitoring the lock management subsystem in a cluster environment. The class consists of the following data items:

- New ENQ Rate (Local)—Rate of new lock (ENQ) requests that originate and are performed on this system

- New ENQ Rate (Incoming)—Rate of new lock requests that originate on other systems and are performed on this system

- New ENQ Rate (Outgoing)—Rate of new lock requests that originate on this system and are performed on another system

- Converted ENQ Rate (Local)—Rate of lock (ENQ) conversion requests that originate and are performed on this system

- Converted ENQ Rate (Incoming)—Rate of lock conversion requests that originate on other systems and are performed on this system

- Converted ENQ Rate (Outgoing)—Rate of lock conversion requests that originate on this system and are performed on another system

- DEQ Rate (Local)—Rate of unlock (DEQ) requests that originate and are performed on this system

- DEQ Rate (Incoming)—Rate of unlock requests that originate on other systems and are performed on this system

- DEQ Rate (Outgoing)—Rate of unlock requests that originate on this system and are performed on another system

- Blocking AST Rate (Local)—Rate of lock manager blocking ASTs that originate and are performed on this system

- Blocking AST Rate (Incoming)—Rate of lock manager blocking ASTs that originate on other systems and are performed on this system

- Blocking AST Rate (Outgoing)—Rate of lock manager blocking ASTs that originate on this system and are performed on another system

- Directory Function Rate (Incoming)—Rate of requests for locks being managed by this node

- Directory Function Rate (Outgoing)—Rate of requests for locks being managed by other nodes

- Deadlock Message Rate—Rate of incoming and outgoing messages required for deadlock detection

---

EXAMPLE

```
MONITOR> MONITOR DLOCK
                        VAX/VMS Monitor Utility
                 DISTRIBUTED LOCK MANAGEMENT STATISTICS
                           on node SAMPLE
                      31-DEC-1988 11:02:20

                                  CUR       AVE       MIN       MAX

New ENQ Rate         (Local)     15.84     11.59      1.54     26.88
                   (Incoming)     1.67      2.62      0.11     25.05
                   (Outgoing)     0.05      0.63      0.00      5.99
Converted ENQ Rate   (Local)     23.67      9.13      0.99     41.22
                   (Incoming)     4.48      5.71      0.00     70.19
                   (Outgoing)     0.00      1.43      0.00     15.90
DEQ Rate             (Local)     15.86     11.58      1.64     26.68
                   (Incoming)     1.66      2.59      0.00     24.85
                   (Outgoing)     0.05      0.63      0.00      5.99
Blocking AST Rate    (Local)      0.00      0.00      0.00      0.01
                   (Incoming)     0.00      0.00      0.00      0.00
                   (Outgoing)     0.00      0.00      0.00      0.00
Dir Functn Rate    (Incoming)     8.00      7.33      4.66     11.00
                   (Outgoing)     1.00      0.77      0.00      2.66
Deadlock Message Rate             0.00      0.00      0.00      0.00
```

This example shows that most of the current lock management activity occurs locally, but that, at some point during the monitoring period, there was a significant amount of incoming activity.

MONITOR FCP

The MONITOR FCP command initiates monitoring of the File Control Primitive statistics class, which includes information on all Files–11 Ancillary Control Processes (ACPs) and Extended QIO Programs (XQPs) on the local node.

FORMAT **MONITOR FCP**

QUALIFIERS */qualifier[,...]*
One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL
Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE
Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT
Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM
Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM
Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION The FCP class consists of the following data items, all of which are displayed as occurrences per second:

- FCP Call Rate—Rate of QIO requests received by the file system.

- Allocation Rate—Rate of calls that caused allocation of disk space.

- Create Rate—Rate at which new files were created.

- Disk Read Rate—Rate of read I/O operations from disk by the file system.

- Disk Write Rate—Rate of write I/O operations to disk by the file system.

- Volume Lock Wait Rate—Rate of entry into a wait state due to contention for a volume synchronization lock. Volume synchronization locks are taken out by the XQP during file creation, deletion, extension, and truncation operations.

- CPU Tick Rate—Rate at which CPU time was used by the file system (in 10-millisecond ticks).

- File System Page Fault Rate—Rate at which page faults occurred in the file system.

- Window Turn Rate—Rate of file-map window misses.

- File Lookup Rate—Rate of file name look-up operations in file directories.

- File Open Rate—Rate at which files were opened.

- Erase Rate—Rate of erase operations issued by the file system.

EXAMPLE

```
MONITOR> MONITOR /INTERVAL=10 FCP
                      VAX/VMS Monitor Utility
                    FILE PRIMITIVE STATISTICS
                         on node SAMPLE
                    31-DEC-1988 16:13:38
                            CUR       AVE       MIN       MAX
FCP Call Rate               4.62      3.80      0.33      7.61
Allocation Rate             0.99      0.24      0.00      0.99
Create Rate                 2.31      0.57      0.00      2.31

Disk Read Rate              1.98      2.48      0.33      6.95
Disk Write Rate             3.30      2.39      0.33      5.62
Volume Lock Wait Rate       4.62      3.06      0.00      6.95

CPU Tick Rate               3.63      3.88      0.33     10.26
File Sys Page Fault Rate    0.00      0.00      0.00      0.00
Window Turn Rate            1.98      0.99      0.00      1.98

File Lookup Rate            0.33      1.40      0.00      4.63
File Open Rate              2.00      3.54      2.00      5.10
Erase Rate                  0.00      0.00      0.00      0.00
```

This example shows that the rate of file opens during the last 10-second collection interval was 2.0 (for a total of 20). The average rate since the MONITOR command was entered is 3.54; the highest rate achieved during any 10-second interval is 5.10, and the lowest rate of 2.0 occurred during the last interval.

MONITOR FILE_SYSTEM_CACHE

The MONITOR FILE_SYSTEM_CACHE command initiates monitoring of the FILE_SYSTEM_CACHE statistics class.

FORMAT	**MONITOR FILE_SYSTEM_CACHE**

QUALIFIERS	*/qualifier[,...]* One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL

Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE

Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT

Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM

Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM

Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION

The FILE_SYSTEM_CACHE class includes the following data items:

- Directory FCB Hit%—Percentage of directory file control block hits on the Directory Cache. The percentage value shown is the ratio of hits to the sum of hits plus misses.

- Directory FCB Attempt Rate—Rate at which attempts were made to find directory file control blocks in the Directory Cache.

- Directory Data Hit%—Percentage of directory data hits on the Directory Cache. The percentage value shown is the ratio of hits to the sum of hits plus misses.

- Directory Data Attempt Rate—Rate at which attempts were made to find directory data in the Directory Cache.

- File Header Hit%—Percentage of file header hits on the File Header Cache. The percentage value shown is the ratio of hits to the sum of hits plus misses.

- File Header Attempt Rate—Rate at which attempts were made to find file headers in the File Header Cache.

- File ID Hit%—Percentage of file identifier hits on the File ID Cache. The percentage value shown is the ratio of hits to the sum of hits plus misses.

- File ID Cache Attempt Rate—Rate at which attempts were made to find file identifiers in the File ID Cache.

- Extent Cache Hit%—Percentage of appropriate size extent hits on the Extent Cache. The percentage value shown is the ratio of hits to the sum of hits plus misses.

- Extent Cache Attempt Rate—Rate at which attempts were made to find appropriate size extents in the Extent Cache.

- Quota Cache Hit%—Percentage of quota entry hits on the Quota Cache. The percentage value shown is the ratio of hits to the sum of hits plus misses.

- Quota Cache Attempt Rate—Rate at which attempts were made to find entries in the Quota Cache.

- Bitmap Cache Hit%—Percentage of entry hits on the Bitmap Cache. The percentage value shown is the ratio of hits to the sum of hits plus misses.

- Bitmap Cache Attempt Rate—Rate at which attempts were made to find entries in the Bitmap Cache.

Note that all items shown in the FILE_SYSTEM_CACHE display except Dir FCB apply only to XQPs. The Dir FCB item applies to both XQPs and the ODS-1 ACP.

EXAMPLE

```
MONITOR> MONITOR FILE_SYSTEM_CACHE
                        VAX/VMS Monitor Utility
                     FILE SYSTEM CACHING STATISTICS
                           on node SAMPLE
                        31-DEC-1988 13:08:53
```

		CUR	AVE	MIN	MAX
Dir FCB	(Hit %)	100.00	100.00	0.00	100.00
	(Attempt Rate)	1.66	0.49	0.00	1.66
Dir Data	(Hit %)	100.00	100.00	0.00	100.00
	(Attempt Rate)	4.66	1.24	0.00	4.66
File Hdr	(Hit %)	66.00	80.00	0.00	100.00
	(Attempt Rate)	1.00	0.41	0.00	1.00
File ID	(Hit %)	0.00	0.00	0.00	0.00
	(Attempt Rate)	0.00	0.00	0.00	0.00
Extent	(Hit %)	0.00	100.00	0.00	100.00
	(Attempt Rate)	0.00	0.24	0.00	1.00
Quota	(Hit %)	0.00	100.00	0.00	100.00
	(Attempt Rate)	0.00	0.16	0.00	0.66
Bitmap	(Hit %)	0.00	0.00	0.00	0.00
	(Attempt Rate)	0.00	0.00	0.00	0.00

The cache hits and misses reflect the effectiveness of file system caching. Generally, the size of the cache affects the hit rate. The Attempt Rate is the sum of hits plus misses; the Hit% is the percentage of attempts that were successful.

Unlike other MONITOR data items, the averages for the hit percentages are not calculated based on previous hit percentages. Instead, these values are calculated based on the total number of hits and the total number of attempts on a cache since the beginning of the Monitor request. This provides more accurate average values for the hit percentage items.

The *Directory FCB Cache* is checked whenever a directory lookup is performed. Directory lookups can be performed on file open, creation, deletion, extension, or truncation. If the file control block associated with the directory is found in the cache, a hit is recorded. Otherwise a miss is recorded. Both hits and misses are counted as attempts.

The *Directory Data Cache* is checked whenever a file lookup is performed. Directory lookups may be performed on file open, creation, deletion, extension, or truncation. If an entry for the file being accessed is found in the directory data cache, a hit is recorded. Otherwise a miss is recorded. Both hits and misses are counted as attempts.

The *File Header Cache* is checked on file open, close, creation, deletion, extension, or truncation. If the file header for the file being accessed is found in the File Header Cache, a hit is recorded. Otherwise a miss is recorded. Both hits and misses are counted as attempts.

The *File Identification Cache* is a list of file identifiers that are removed on file creation and returned on file deletion. The File Id hits indicate file numbers successfully removed or returned to the File Id Cache. Otherwise a miss is recorded. Both hits and misses are counted as attempts.

The *Extent Cache* is checked on file creation, deletion, extension, and truncation. An attempt is made to allocate space from the Extent Cache during file creation or extension. During file creation, if sufficient size is found, a hit is recorded. If the desired size is not found, or an entry is forced to be split, an attempt is recorded. During file deletion, if the blocks were returned to the cache without the Extent Cache becoming too large, a hit is recorded. Otherwise a miss is recorded. Both hits and misses are counted as attempts.

If quota checking is enabled, the *Quota Cache* is checked on file creation, deletion, extension, and truncation. If the desired entry (the identifier matching that of the requester) is found in the Quota Cache, a hit is recorded. Otherwise a miss is recorded. Both hits and misses are counted as attempts.

The *Bitmap Cache* contains blocks from the storage bitmap file. This cache is accessed when the Extent Cache cannot satisfy requests for disk space. High rates indicate fragmented volumes.

Data items in the FILE_SYSTEM_CACHE display correspond to SYSGEN ACP/XQP parameters, as follows:

FILE_SYSTEM_CACHE Item	ACP/XQP Parameters
Dir FCB	ACP_SYSACC
	ACP_DINDXCACHE
Dir Data	ACP_DIRCACHE
File Hdr	ACP_HDRCACHE
File ID	ACP_FIDCACHE
Extent	ACP_EXTCACHE
	ACP_EXTLIMIT
Quota	ACP_QUOCACHE
Bitmap	ACP_MAPCACHE

When you change the ACP/XQP cache parameters, remember to reboot the system to make the changes effective. For more information on these parameters, refer to the *VMS System Generation Utility Manual*.

MONITOR IO

The MONITOR IO command initiates monitoring of the I/O class.

FORMAT	**MONITOR IO**

QUALIFIERS	**/qualifier[,...]** One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL

Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE

Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT

Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM

Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM

Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION

The IO class includes the following data items:

- Direct I/O Rate—Rate of direct I/O (for example, disk and tape) operations

- Buffered I/O Rate—Rate of buffered I/O (for example, terminal and line printer) operations

- Mailbox Write Rate—Rate of write-to-mailbox requests received by the system

- Split Transfer Rate—Rate at which transfers were split into multiple I/Os.

- Log Name Translation Rate—Rate of logical name translations

- File Open Rate—Rate at which files were opened

- Page Fault Rate—Rate of occurrence of page faults for all working sets

- Page Read Rate—Rate of pages read from disk as a result of page faults

- Page Read I/O Rate—Rate of read I/O operations from disk as a result of page faults

- Page Write Rate—Rate of pages written to the page file

- Page Write I/O Rate—Rate of write I/O operations to the page file

- Inswap Rate—Rate at which working sets were read into memory from the swap file

- Free List Size—Number of pages on the free page list

- Modified List Size—Number of pages on the modified page list

EXAMPLE

```
MONITOR> MONITOR /RECORD IO
```

```
                        VAX/VMS Monitor Utility
                        I/O SYSTEM STATISTICS
                           on node SAMPLE
                        31-DEC-1988 22:22:44
```

	CUR	AVE	MIN	MAX
Direct I/O Rate	15.33	4.46	0.33	15.33
Buffered I/O Rate	24.91	47.47	24.91	69.00
Mailbox Write Rate	0.00	0.45	0.00	2.95
Split Transfer Rate	1.66	1.56	0.33	3.97
Log Name Translation Rate	13.28	10.75	3.66	27.66
File Open Rate	1.66	1.26	0.33	2.98
Page Fault Rate	24.58	52.31	17.33	178.00
Page Read Rate	12.29	9.00	0.00	26.88
Page Read I/O Rate	2.65	2.43	0.00	6.22
Page Write Rate	0.00	6.69	0.00	58.66
Page Write I/O Rate	0.00	0.27	0.00	1.66
Inswap Rate	0.00	0.00	0.00	0.00
Free List Size	3621.00	3604.09	3392.00	3771.00
Modified List Size	49.00	73.36	4.00	181.00

                                                          RECORDING

This example shows that the direct I/O rate is currently at its highest level since the MONITOR command was entered and is significantly higher than the average rate. Termination of this command by CTRL/C and entry of a MONITOR PROCESSES/TOPDIO command would show the top users of direct I/Os. Note that if I/O monitoring is begun at a later time, a new MONITOR request is defined. That is, it is not a continuation of the original request; the average, minimum, and maximum statistics are reinitialized. However, since the original request specified recording, that data can be played back for redisplay or summarization.

MONITOR LOCK

The MONITOR LOCK command initiates monitoring of the LOCK class.

FORMAT	**MONITOR LOCK**

QUALIFIERS	*/qualifier[,...]* One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL
Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE
Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT
Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM
Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM
Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION

The LOCK class includes the following data items:

- New ENQ Rate—Rate of new lock (ENQ) requests (as opposed to conversions)

- Converted ENQ Rate—Rate of lock (ENQ) conversion requests

- DEQ Rate—Rate of unlock (DEQ) requests

- Blocking AST Rate—Rate of lock manager blocking ASTs delivered

- ENQs Forced To Wait Rate—Rate of occurrence of locks that could not be granted immediately, thus having to wait

- ENQs Not Queued Rate—Rate of occurrence of locks that could not be granted immediately but requested not to be queued, and thus received an error status instead

- Deadlock Search Rate—Rate at which a deadlock search was performed

- Deadlock Find Rate—Rate at which a deadlock was found

- Total Locks—Total number of locks in the system

- Total Resources—Total number of resources in the system

EXAMPLE

```
MONITOR> MONITOR /INPUT=LOCKSTATS.DAT/SUMMARY/NODISPLAY LOCK/AVERAGE
        .
        .
        .
MONITOR> CTRL/Z
$ TYPE MONITOR.SUM

                            VAX/VMS Monitor Utility
        +-----+         LOCK MANAGEMENT STATISTICS
        | AVE |             on node SAMPLE          From:   31-DEC-1988 08:00:00
        +-----+                SUMMARY              To:     31-DEC-1988 17:00:00

                            0         5        10        15        20
                            + - - - - + - - - - + - - - - + - - - - -+
New ENQ Rate                2 |****
Converted ENQ Rate          1 |**
                            |         |         |         |         |
DEQ Rate                    3 |******
Blocking AST Rate           |
                            |         |         |         |         |
ENQs Forced To Wait Rate    |
ENQs Not Queued Rate        |
                            |         |         |         |         |
Deadlock Search Rate        |
Deadlock Find Rate          |
                            |         |         |         |         |
Total Locks                 3 |******
Total Resources             3 |******
                            |         |         |         |         |
                            + - - - - + - - - - + - - - - + - - - - -+
PLAYBACK                         SUMMARIZING
```

This example shows the average use of the lock management subsystem during a typical workday, based on data that was previously recorded.

MONITOR MODES

The MONITOR MODES command initiates monitoring of the MODES class, which includes a data item for each mode of processor operation.

FORMAT   **MONITOR MODES**

QUALIFIERS   */qualifier[,...]*
One or more qualifiers as described in the Qualifiers section.

CLASS-NAME   */ALL*
QUALIFIERS   Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE
Selects average statistics to be displayed in a bar graph for display and summary output.

/[NO]CPU [=(x[,...])]
In multiprocessor configurations, selects the CPU-specific form of output, where x specifies the CPU identification. If you specify /CPU without specifying a CPU identification, MONITOR displays MODES class statistics for each successive CPU until information for all active CPUs has been displayed. MONITOR then repeats the cycle beginning with the first CPU. If you specify one CPU identification, MONITOR displays statistics for that CPU only. If you specify multiple CPU identifications, MONITOR displays statistics for each successive CPU specified, then repeats the cycle beginning with the first specified CPU.

The default is /NOCPU. For multiprocessor systems, /NOCPU produces a single modes screen that reflects the combined time that all CPUs spent in each mode.

The /CPU qualifier has no effect on output for nonmultiprocessor systems.

/CURRENT
Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM
Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM
Selects minimum statistics to be displayed in a bar graph for display and summary output.

/[NO]PERCENT

Controls whether statistics are expressed as percent values in display and summary output. The /PERCENT qualifier is applicable only to the DISK, MODES, SCS, and STATES classes. By default, statistics are not expressed as percent values.

DESCRIPTION

The following data items, included by the MODES class, can be displayed as percentages of all processor (CPU) time or as rates of clock ticks (10 millisecond units) per second:

- Interrupt Stack—Time spent on the interrupt stack

- MP Synchronization—Time spent synchronizing multiple CPUs (applicable to multiprocessor systems only)

- Kernel Mode—Time spent in kernel mode, but not on interrupt stack

- Executive Mode—Time spent in executive mode

- Supervisor Mode—Time spent in supervisor mode

- User Mode—Time spent in user mode executing VAX instructions

- Compatibility Mode—Time spent executing compatibility mode instructions

- Idle Time—Time spent executing the NULL process

For multiprocessor systems, when you enter the MONITOR MODES command without using the /CPU qualifier to select specific CPUs, MONITOR produces a single modes screen similar to those produced for non-multiprocessor systems. However, the statistics produced for multiprocessor systems reflect the combined time that all CPUs spent in each mode.

EXAMPLES

1   MONITOR> MONITOR MODES /PERCENT

```
                              VAX/VMS Monitor Utility
                              TIME IN PROCESSOR MODES (%)
               +-----+              on node SAMPLE
               | CUR |          31-DEC-1988  22:52:42
               +-----+
                              0%      25%     50%     75%     100%
                              + - - - + - - - + - - - + - - - -+
   Interrupt Stack         4 |*
                             |       |       |       |        |
   MP Synchronization        |
                             |       |       |       |        |
   Kernel Mode             6 |**
                             |       |       |       |        |
   Executive Mode          2 |
                             |       |       |       |        |
   Supervisor Mode           |
                             |       |       |       |        |
   User Mode              72 |**************************
                             |       |       |       |        |
   Compatibility Mode        |
                             |       |       |       |        |
   Idle Time              16 |******
                             |       |       |       |        |
                              + - - - + - - - + - - - + - - - -+
```
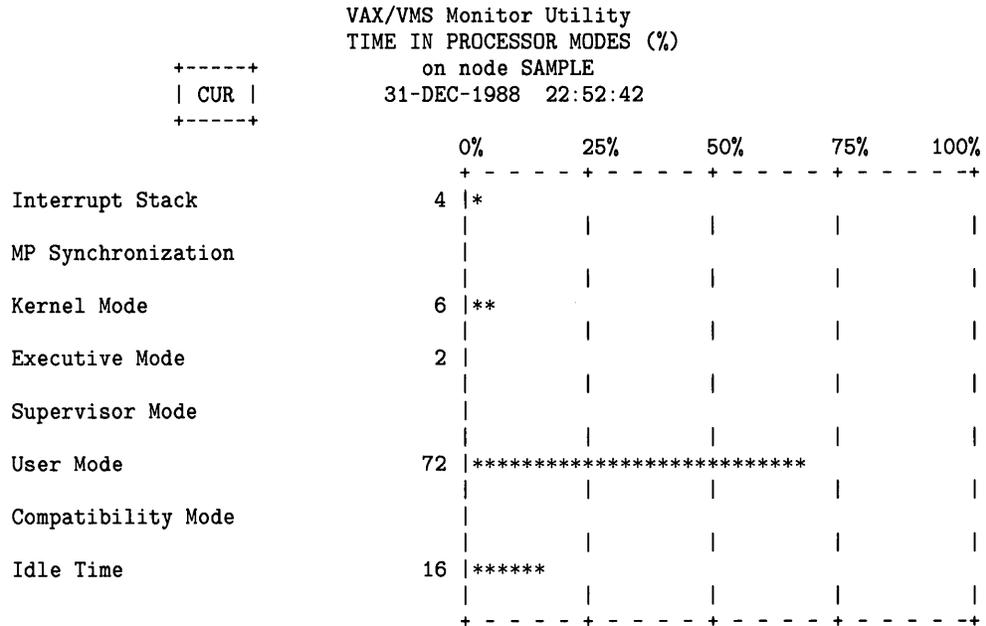
This display shows that, over the last collection interval, the processor spent 72% of its time executing user code, 8% executing system code to service user requests in executive and kernel modes, and 4% processing interrupts on the interrupt stack; it was idle 16% of the time. Time spent executing VAX RMS code is included in executive mode time. Time spent executing DCL code is included in supervisor mode time. The majority of interrupt stack time is devoted to processing buffered I/O requests.

If you omit the /PERCENT qualifier or specify /NOPERCENT, MONITOR displays mode times as rates of clock ticks per second, where a clock tick is 10 milliseconds.

❷   MONITOR> MONITOR MODES

```
                                    VAX/VMS Monitor Utility
                +------+            TIME IN PROCESSOR MODES
                | CUR  |                on node SAMPLE
                +------+            31-DEC-1988 15:02:36

Combined for  2 CPUs                  0 .       50        100       150       200
                                      + - - - - + - - - - + - - - - + - - - - -+
  Interrupt Stack                     |
                                      |         |         |         |         |
  MP Synchronization                  |
                                      |         |         |         |         |
  Kernel Mode                       2 |*
                                      |         |         |         |         |
  Executive Mode                    1 |*
                                      |         |         |         |         |
  Supervisor Mode                     |
                                      |         |         |         |         |
  User Mode                       101 |*******************
                                      |         |         |         |         |
  Compatibility Mode                  |
                                      |         |         |         |         |
  Idle Time                        96 |*****************
                                      + - - - - + - - - - + - - - - + - - - - -+
```

This example demonstrates output of the MONITOR MODES command for
a multiprocessor system. Displayed statistics represent rates of clock ticks per
second. Information in the upper left corner of the screen indicates that node
QUATRE has four CPUs, two of which are active. Since the command line
does not include the /CPU qualifier, statistics reflect the combined time that
all CPUs spent in each mode.

MONITOR MSCP_SERVER

The MONITOR MSCP_SERVER command initiates monitoring of the Mass Storage Control Protocol (MSCP) Server class.

FORMAT	**MONITOR MSCP_SERVER**

QUALIFIERS	**/qualifier[,...]** One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL

Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE

Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT

Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM

Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM

Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION

The MSCP Server class consists of several data items that can be useful in tuning the MSCP Server. These data items are:

* Server I/O Request Rate—The rate at which remote processors request I/O transfers.

* Read Request Rate—The rate at which remote processors request Read I/O transfers.

* Write Request Rate—The rate at which remote processors request Write I/O transfers.

- Extra Fragment Rate—The rate at which the server issues extra fragments. One or more extra fragments are created when, due to buffering constraints, the MSCP Server issues multiple I/Os in order to fulfill a single I/O request. For example, if the MSCP server breaks up a 64 block request into 4 fragments of 16 blocks, three extra fragments are created.

- Fragmented Request Rate—The rate at which fragmented requests occur. A fragmented request is a transfer request that the server fragments due to buffering constraints. For example, one fragmented request occurs when the server splits a 36 block request into three fragments of sizes 16 blocks, 16 blocks, and 4 blocks. In this example, the server creates two extra fragments.

- Buffer Wait Rate—The rate at which "buffer waits" occur in the server. A buffer wait occurs when a request must wait for MSCP buffer memory.

- Request Size Rates—A histogram that displays the rate of requests for various block sizes.

EXAMPLE

```
MONITOR> MONITOR MSCP_SERVER
```

```
                    VAX/VMS Monitor Utility
                    MSCP SERVER STATISTICS
                       on node GLOBBO
                    31-DEC-1988 09:51:43
```

		CUR	AVE	MIN	MAX
Server I/O Request Rate		0.00	0.71	0.00	6.22
Read Request Rate		0.00	0.54	0.00	6.22
Write Request Rate		0.00	0.16	0.00	6.16
Extra Fragment Rate		0.00	0.00	0.00	0.00
Fragmented Request Rate		0.00	0.00	0.00	0.00
Buffer Wait Rate		0.00	0.00	0.00	0.00
Request Size Rates	1	0.00	0.07	0.00	0.98
(Blocks)	2-3	0.00	0.03	0.00	0.65
	4-7	0.00	0.03	0.00	0.65
	8-15	0.00	0.10	0.00	1.63
	16-31	0.00	0.46	0.00	5.51
	32-63	0.00	0.00	0.00	0.00
	64+	0.00	0.00	0.00	0.00

This example demonstrates use of the MONITOR MSCP_SERVER command to generate MSCP statistics on node GLOBBO.

MONITOR PAGE

The MONITOR PAGE command initiates monitoring of the PAGE class.

FORMAT	**MONITOR PAGE**

QUALIFIERS	*/qualifier[,...]* One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL

Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE

Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT

Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM

Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM

Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION

The PAGE class includes the following data items:

- Page Fault Rate—Rate of page faults for all working sets

- Page Read Rate—Rate of pages read from disk as a result of page faults

- Page Read I/O Rate—Rate of read I/O operations from disk as a result of page faults

- Page Write Rate—Rate at which pages were written to the page file

- Page Write I/O Rate—Rate of write I/O operations to the page file

- Free List Fault Rate—Rate at which pages were read from the free page list as a result of page faults

- Modified List Fault Rate—Rate of pages read from the modified page list as a result of page faults

- Demand Zero Fault Rate—Rate at which zero-filled pages were allocated as a result of page faults

- Global Valid Fault Rate—Rate of page faults for pages that are not in the process's working set, but are in physical memory and are indicated as valid pages in the systemwide global page tables

- Writes In Progress Fault Rate—Rate of pages read that were in the process of being written back to disk when faulted

- System Fault Rate—Rate of page faults for pages in system space

- Free List Size—Number of pages on the free page list

- Modified List Size—Number of pages on the modified page list

EXAMPLE

MONITOR> MONITOR PAGE

```
                        VAX/VMS Monitor Utility
                       PAGE MANAGEMENT STATISTICS
                            on node SAMPLE
                        31-DEC-1988 22:22:44

                              CUR       AVE       MIN       MAX

Page Fault Rate              26.82     18.27      9.66     26.82
Page Read Rate                3.97      2.65      1.33      3.97
Page Read I/O Rate            1.32      0.99      0.66      1.32
Page Write Rate               0.00      0.00      0.00      0.00
Page Write I/O Rate           0.00      0.00      0.00      0.00

Free List Fault Rate         13.90     10.96      8.00     13.90
Modified List Fault Rate      5.62      2.99      0.33      5.62
Demand Zero Fault Rate        4.63      2.65      0.66      4.63
Global Valid Fault Rate       1.32      0.66      0.00      1.32
Wrt In Progress Fault Rate    0.00      0.00      0.00      0.00
System Fault Rate             2.31      1.99      1.66      2.31

Free List Size             3164.00   3176.00   3164.00   3188.00
Modified List Size          155.00    131.00    107.00    155.00
```

This example shows that the current rate of pages read per read I/O operation is approximately 3 per second (Page Read Rate divided by Page Read I/O Rate). Note that while the page fault rate is currently at the highest point of the monitoring session, the majority of the pages are faulted from memory, not from disk.

MONITOR POOL

The MONITOR POOL command initiates monitoring of the POOL class, which measures space allocations in the nonpaged dynamic pool.

FORMAT	**MONITOR POOL**

QUALIFIERS	***/qualifier[,...]*** One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL

Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE

Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT

Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM

Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM

Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION

The POOL class includes the following data items:

- SRPs Available—Number of small request packets available in the SRP queue

- SRPs In Use—Number of small request packets available in use

- IRPs Available—Number of intermediate request packets available in the IRP queue

- IRPs In Use—Number of intermediate request packets available in use

- LRPs Available—Number of large request packets available in the LRP queue

- LRPs In Use—Number of large request packets in use

- Dynamic Kilobytes Available—Number of dynamic kilobytes available

- Dynamic Kilobytes In Use—Number of dynamic kilobytes in use

- Holes In Pool—Unused blocks of contiguous space in the dynamically allocated portion of the pool

- Largest Block—Size in kilobytes of the largest block of unused space in the dynamically allocated portion of the pool

- Smallest Block—Size in bytes of the smallest block of unused space in the dynamically allocated portion of the pool

- Blocks Less or Equal 32 Bytes—Blocks less than or equal to 32 bytes in size in the dynamically allocated portion of the pool

EXAMPLE

```
MONITOR> MONITOR/RECORD POOL/MINIMUM

                              VAX/VMS Monitor Utility
              +-----+         NONPAGED POOL STATISTICS
              | MIN |             on node SAMPLE
              +-----+         31-DEC-1988 11:29:09

                                0K       3K       6K       9K      12K
                              /
                              + - - - + - - - + - - - + - - - - +
SRPs Available             1 |
SRPs In Use            10345 |********************************
                             |        |        |        |        |
IRPs Available            69 |
IRPs In Use             2520 |********
                             |        |        |        |        |
LRPs Available            66 |
LRPs In Use               60 |
                             |        |        |        |        |
Dynamic Kbytes Available  87 |
Dynamic Kbytes In Use   1659 |*****
Holes In Pool            712 |**
Largest Block (Kbytes)    15 |
Smallest Block (Bytes)    16 |
Blocks Less or Eq 32 Bytes 317 |*
                             + - - - + - - - + - - - + - - - - +
                                                        RECORDING
```

In this example, only the minimum statistic is being displayed, but data is being recorded that could later be used to redisplay all statistics. Of particular interest in this display are the SRPs, IRPs, and LRPs Available items. If queue sizes for these preallocated packets drop to 0 at any time, extra overhead is incurred by the subsystems requiring these packets. Initial queue sizes of these items are determined by system parameters that can be adjusted at boot time.

MONITOR PROCESSES

The MONITOR PROCESSES command initiates monitoring of the PROCESSES class, which displays information on all processes in the system.

FORMAT	**MONITOR PROCESSES**

QUALIFIERS	*/qualifier[,...]* One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS	*/TOPBIO* Specifies that a bar graph listing the top buffered I/O users be produced instead of the standard display and summary output. Values are expressed in units of buffered I/Os per second. */TOPCPU* Specifies that a bar graph listing the top CPU time users be produced instead of the standard display and summary output. Values are expressed in units of clock ticks (10 milliseconds) per second. */TOPDIO* Specifies that a bar graph listing the top direct I/O users be produced instead of the standard display and summary output. Values are expressed in units of direct I/Os per second. */TOPFAULT* Specifies that a bar graph listing the top page faulting processes be produced instead of the standard display and summary output. Values are expressed in units of page faults per second.

DESCRIPTION	As illustrated in the examples, the PROCESSES display (and summary) formats are different from those of all other classes. The PROCESSES display provides the following information:

- PID—Process identification as assigned by the system, in hexadecimal

- STATE—Process's scheduler state (see the description of the MONITOR STATES command for an explanation of the STATES codes)

- PRI—Current (as opposed to base) priority of the process

- NAME—Process name

- PAGES—Number of shareable pages and total number of pages currently in use by the process

- DIOCNT—Cumulative direct I/O operations performed by the process since its creation; not displayed if the process is swapped out

- FAULTS—Cumulative page faults since the process was created; not displayed if the process is swapped out

- CPU TIME—Cumulative CPU time used by the process since its creation, in the format *hours:minutes:seconds*; not displayed if the process is swapped out

The top corners of the display contain the number of processes in the system and the time in days, hours, minutes, and seconds since the system was last booted. Processes that are swapped out are so noted.

If more processes are in the system than can be displayed on the terminal screen at once, the display consists of multiple screens. Screens are presented one at a time at intervals specified with the /VIEWING_TIME qualifier. Note that the four /TOP bar graph displays provide the PID and process name of each of the top eight users.

As with the other bar graph displays, examples in the displays of top users are rounded to the nearest whole number. Up to eight processes with nonzero values are displayed. To be eligible for inclusion in the list of top users, a process must be present and swapped in at the beginning and end of the display interval. This eligibility requirement also applies to the beginning and ending of the entire period covered by a summary.

Note that only one of the displays of top users or the regular PROCESSES display can be selected in a single MONITOR request.

EXAMPLES

1 MONITOR> MONITOR/INPUT=PROCS.DAT/INTERVAL=6 PROCESSES

```
Process Count: 20              VAX/VMS Monitor Utility        Uptime:     1 23:26:10
                                      PROCESSES
                                     on node SAMPLE
                                 31-DEC-1988 12:39:09

       PID    STATE PRI    NAME         PAGES       DIOCNT  FAULTS  CPU TIME

    00000080 COM    0 NULL          0/0              0        0 19:43:13.8
    00000081 HIB   16 SWAPPER       0/0              0        0 00:00:15.8
    00000102 LEFO   4 SAMPLE1001    87/232      SWAPPED OUT
    00000103 COM    4 SAMPLE1101    16/100        7127    51298 00:05:11.0
    00000084 HIB    8 ERRFMT        64/174        2750      125 00:00:43.9
    00000086 LEF    8 OPCOM         73/272         283      178 00:00:07.7
    00000087 HIB    9 JOB_CONTROL   57/293         707      167 00:00:10.5
    00000088 HIB    8 CONFIGURE     43/205          22      123 00:00:00.6
    0000008A HIB    6 SYMBIONT_0001 5/56            50      617 00:03:15.1
    0000008B HIB    8 JNLACP        75/580       15149     4922 00:21:51.1
    0000008C HIB    8 NETACP        5/954           11     1057 00:25:06.8
    0000008D HIB    5 EVL           7/56            44    34384 00:00:20.5
    0000008E HIB    9 REMACP        5/54            13      107 00:00:01.3
    00000112 COM    4 SAMPLE1601    45/111       13131    39992 00:06:39.1
    0000011E CUR    9 SMITH         89/298         138      830 00:00:07.1

PLAYBACK
```

This example illustrates a PROCESSES display generated from the input file PROCS.DAT. One line is displayed for each process in the system. Note that this display shows current values only—average, minimum, and maximum statistics are not available. Also note that for swapped-out processes, the words SWAPPED OUT replace the three rightmost items, because those items

are not available for swapped-out processes. Since this example is a playback request, the system uptime displayed is that of the system at the time the MONITOR data was recorded.

Nondisplayable characters in process names are represented by periods.

2 MONITOR> MONITOR/INPUT=PROCS.DAT PROCESSES/TOPDIO

```
                          VAX/VMS Monitor Utility
                      TOP DIRECT I/O RATE PROCESSES
                             on node SAMPLE
                         31-DEC-1988 16:13:38

                                   0        25       50       75       100
                                   + - - - + - - - + - - - + - - - + - - -+
    000000C7  SAMPLE0901      25   |*********
                                   |         |        |        |        |
    00000112  SAMPLE1601      17   |******
                                   |         |        |        |        |
    00000102  SAMPLE1001      14   |*****
                                   |         |        |        |        |
    00000103  SAMPLE1101      12   |****
                                   |         |        |        |        |
    00000080  NULL            12   |****
                                   |         |        |        |        |
    0000011E  SMITH            4   |*
                                   |         |        |        |        |
    0000008C  NETACP           1   |
                                   |         |        |        |        |
                                   |
                                   + - - - + - - - + - - - + - - - + - - -+
```

PLAYBACK

This example shows that the process SAMPLE0901, with a rate of 25 per second, was the top consumer of direct I/Os during the most recent interval between displays.

MONITOR RMS

The MONITOR RMS command initiates monitoring of the VAX Record Management Services (RMS) statistics class for a specific file.

FORMAT   **MONITOR RMS**

QUALIFIERS   */qualifier[,...]*
One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL
Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE
Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT
Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/FILE=(file-name[,...])
Specifies a list of one or more files to which the MONITOR RMS command applies. If you include a node name as part of the file specification, MONITOR ignores the node name. Use the /NODE command qualifier to select specific nodes for MONITOR RMS requests. If you use the /NODE command qualifier to specify multiple nodes, the file must exist on all specified nodes. You can list up to 5,000 files. Do not specify wildcard characters.

/ITEM=(keyword[,...])
Selects one or more data items for inclusion in display and summary output. If you specify two or more keywords, enclose them in parentheses, and separate them with commas. When the /ITEM qualifier is omitted, the default is /ITEM=OPERATIONS.

/MAXIMUM
Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM
Selects minimum statistics to be displayed in a bar graph for display and summary output.

/ITEM QUALIFIER KEYWORDS	**OPERATIONS** Specifies that RMS basic operations statistics are displayed for the selected file. **DATA_RATES** Specifies that RMS data rate statistics are displayed for the selected file. **LOCKING** Specifies that RMS locking statistics are displayed for the selected file. **CACHING** Specifies that RMS caching statistics are displayed for the selected file.

DESCRIPTION

When you enter the MONITOR RMS command, you must use the /FILE qualifier to specify an input file. MONITOR displays RMS statistics for the input file that you specify. MONITOR only displays statistics for the input file if statistics is enabled for the file, and the file is open. For information about enabling statistics for a file, see the SET FILE command in the *VMS DCL Dictionary* and the *VMS Record Management Services Manual*.

The MONITOR RMS command generates RMS statistics of the following types:

- Basic operations (produced by specifying the OPERATIONS item)

- Data rates per operation (produced by specifying the DATA_RATES item)

- File locking (produced by specifying the /LOCKING item)

- Caching (produced by specifying the /CACHING item)

Basic operations statistics consist of the following data items:

- Sequential $Get Call Rate

- Keyed $Get Call Rate

- RFA $Get Call Rate

- Sequential $Find Call Rate

- Keyed $Find Call Rate

- RFA $Find Call Rate

- Sequential $Put Call Rate

- Keyed $Put Call Rate

- $Read Call Rate

- $Write Call Rate

- $Update Call Rate

- $Delete Call Rate

- $Truncate Call Rate

- $Extend Call Rate

- $Flush Call Rate

Data rate statistics consist of the following data items:
- Total $GET Call Rate
- Bytes per $GET
- Total $PUT Call Rate
- Bytes Per $PUT
- Total $UPDATE Call Rate
- Bytes per $UPDATE
- $READ Call Rate
- Bytes per $READ
- $WRITE Call Rate
- Bytes per $WRITE
- $TRUNCATE Call Rate
- Blocks per $TRUNCATE
- $EXTEND Call Rate
- Blocks per $EXTEND

File locking statistics consist of the following data items:
- New ENQ Rate
- DEQ Rate
- Converted ENQ Rate
- Blocking AST Rate
- Bucket Split Rate
- Multi-Bucket Split Rate

Caching statistics consist of the following data items:
- Local Cache Hit Percent
- Local Cache Attempt Rate
- Global Cache Hit Percent
- Global Cache Attempt Rate
- Global Buffer Read I/O Rate
- Global Buffer Write I/O Rate
- Local Buffer Read I/O Rate
- Local Buffer Write I/O Rate

Note: **Values produced by the MONITOR RMS command do not include I/Os generated by the recovery mechanisms of RMS Journaling.**

For more information about RMS, RMS services, and file applications, see the *VMS Record Management Services Manual, VMS System Services Reference Manual*, and the *Guide to VMS File Applications*.

EXAMPLE

```
MONITOR> MONITOR RMS /ITEM=OPERATIONS /FILE=SYS$COMMON:[SYSEXE]SYSUAF.DAT

                        VAX/VMS Monitor Utility
                         RMS FILE OPERATIONS
                            on node SAMPLE
                        31-DEC-1988 11:03:06
(Index)  _$254$DUA213:[SYS0.SYSEXE]SYSUAF.DAT;2
Active Streams:   17                  CUR       AVE       MIN       MAX

     $GET Call Rate    (Seq)         0.00      0.00      0.00      0.00
                       (Key)         4.30      2.15      0.00      6.76
                       (RFA)         0.00      0.00      0.00      0.00
     $FIND Call Rate   (Seq)         0.00      0.00      0.00      0.00
                       (Key)         0.00      0.00      0.00      0.00
                       (RFA)         0.00      0.00      0.00      0.00
     $PUT Call Rate    (Seq)         0.00      0.00      0.00      0.00
                       (Key)         0.20      0.14      0.00      0.30
     $READ Call Rate                 0.00      0.00      0.00      0.00
     $WRITE Call Rate                0.00      0.00      0.00      0.00
     $UPDATE Call Rate               0.00      0.00      0.00      0.00
     $DELETE Call Rate               0.00      0.00      0.00      0.00
     $TRUNCATE Call Rate             0.00      0.00      0.00      0.00
     $EXTEND Call Rate               0.00      0.00      0.00      0.00
     $FLUSH Call Rate                0.00      0.00      0.00      0.00
```

This example demonstrates the use of the MONITOR RMS command to generate basic operations statistics for the file SYSUAF.DAT.

MONITOR SCS

The MONITOR SCS command initiates monitoring of the SCS (System Communication Services) class.

FORMAT	**MONITOR SCS**

QUALIFIERS	**/qualifier[,...]** One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL
Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE
Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT
Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/ITEM=(keyword[,...])
Selects one or more data items for inclusion in display and summary output. If you specify two or more keywords, enclose them in parentheses, and separate them with commas. When the /ITEM qualifier is omitted, the default is /ITEM=KB_MAP.

/MAXIMUM
Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM
Selects minimum statistics to be displayed in a bar graph for display and summary output.

/[NO]PERCENT
Controls whether statistics are expressed as percent values in display and summary output. The /PERCENT qualifier is applicable only to the DISK, MODES, SCS, and STATES classes. By default, statistics are not expressed as percent values.

/ITEM Qualifier Keywords

ALL

Specifies that statistics on all data items collected for the disks are displayed on successive screens.

BUFFER_DESCRIPTOR

Specifies that statistics on the queued-for-buffer-descriptor (on the local node) rate are displayed for each node.

D_DISCARD

Specifies that datagram discard rate statistics are displayed for each node.

D_RECEIVE

Specifies that datagram receive rate statistics are displayed for each node.

D_SEND

Specifies that datagram send rate statistics are displayed for each node.

KB_MAP

Specifies that kilobyte map rate statistics are displayed for each node.

KB_REQUEST

Specifies that kilobyte request (via request datas) rate statistics are displayed for each node.

KB_SEND

Specifies that kilobyte send (via send datas) rate statistics are displayed for each node.

M_RECEIVE

Specifies that message receive rate statistics are displayed for each node.

M_SEND

Specifies that message send rate statistics are displayed for each node.

REQUEST_DATA

Specifies that request data (initiated on the local node) rate statistics are displayed for each node.

SEND_CREDIT

Specifies that queued-for-send-credit (on the local node) rate statistics are displayed for each node.

SEND_DATA

Specifies that send data (initiated on the local node) rate statistics are displayed for each node.

DESCRIPTION

The SCS class is a component class. Data items for this class are collected for each node in the cluster. The SCS class consists of the following data items:

- Datagram Send Rate—Rate at which datagrams are sent to another node.

- Datagram Receive Rate—Rate at which datagrams are received from another node.

- Datagram Discard Rate—Rate at which datagrams are discarded.

- Message Send Rate—Rate at which sequenced messages are sent to another node. Sequenced messages are exchanged between nodes to communicate with Mass Storage Control Protocol (MSCP) disks and the lock manager.

- Message Receive Rate—Rate at which sequenced messages are received from another node. Sequenced messages are exchanged between nodes to communicate with MSCP disks and the lock manager.

- Send Data Rate—Rate at which block send datas are initiated on the local node.

- Kbytes Send Rate—Rate at which kilobytes are sent, as a result of send datas initiated on the local node.

- Request Data Rate—Rate at which request datas are initiated on the local node.

- Kbytes Request Rate—Rate at which kilobytes are received, as a result of request datas initiated on the local node.

- Kbytes Map Rate—Rate at which kilobytes are mapped for block transfers. This is a rough measure of the data transfer rate between the local node and a remote node. Before any transfer can take place, a buffer must be mapped. The size of the accumulated buffers that were mapped is displayed by the Kbytes Map Rate. If request datas or send datas are initiated on the local or a remote node, then the Kbytes Map Rate reflects the number of kilobytes actually transferred between the two nodes.

- Send Credit Queued Rate—Rate at which connections are queued for a send credit. A connection is queued for a send credit whenever all of the buffers that were allocated by the remote node have been used.

- Buffer Descriptor Queued Rate—Rate at which connections are queued for a buffer descriptor. A connection is queued for a buffer descriptor whenever all of the buffer descriptors have been allocated by the local node. You can increase the number of buffer descriptors allocated on the local system by adjusting the system parameter SCSBUFFCNT.

EXAMPLE

```
MONITOR> MONITOR SCS

                        VAX/VMS Monitor Utility
                           SCS STATISTICS
                           on node CURLEY
                        31-DEC-1988 10:21:46

    Kbytes Map Rate              CUR       AVE       MIN       MAX
    CURLEY                       0.00      0.00      0.00      0.00
    MOE                          0.00      0.00      0.00      0.00
    LARRY                        0.00      0.00      0.00      0.00
    SHEMP                        5.64      3.81      1.98      5.64
```

The command in this example requests that kilobyte map rate statistics collected for SCS be displayed for each node in the cluster. The display shows block transfer map activity between the node CURLEY and the Hierarchical Storage Controller (HSC) SHEMP. Note that each node in the cluster is identified by its SCS node name.

MONITOR STATES

The MONITOR STATES command initiates monitoring of the PROCESS STATES class, which shows the number of processes in each of the 14 scheduler states.

FORMAT   **MONITOR STATES**

QUALIFIERS   */qualifier[,...]*
One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS

/ALL
Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE
Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT
Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM
Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM
Selects minimum statistics to be displayed in a bar graph for display and summary output.

/[NO]PERCENT
Controls whether statistics are expressed as percent values in display and summary output. The /PERCENT qualifier is applicable only to the DISK, MODES, SCS, and STATES class names. By default, statistics are not expressed as percent values.

DESCRIPTION   The STATES class shows the number of processes in each of the 14 scheduler states, as follows:

* Collided Page Wait (COLPG)—Waiting for a faulted page in transition.

- Mutex & Miscellaneous Resource Wait (MWAIT)—Waiting for the availability of a mutual exclusion semaphore or a dynamic resource. Following is a summary of Mutex and Miscellaneous Resource Wait states and identifying codes, as they appear in the PROCESSES class display:

MWAIT	Reason for Wait
MUTEX	Mutual exclusion semaphore
RWAST	AST wait (wait for system or special kernel AST)
RWBRK	Breakthrough (wait for broadcast message)
RWCLU	Cluster state transition wait
RWIMG	Image activation lock
RWLCK	Lock data base
RWMBX	Mailbox full
RWMPB	Modified page writer busy
RWMPE	Modified page list empty
RWNPG	Nonpaged dynamic memory
RWPAG	Paged dynamic memory
RWPGF	Page file full
RWQUO	Job quota
RWSCS	System Communication Services wait
RWSWP	Swap file space

- Common Event Flag Wait (CEF)—Waiting for some combination of event flags to be set in a common event block.

- Page Fault Wait (PFW)—Waiting for a page to be read as a result of a page fault; resident processes.

- Local Event Flag Wait (LEF)—Waiting for one or more local event flags to be posted; resident processes.

- Local Event Flag (Outswapped) (LEFO)—Waiting for one or more local event flags to be posted; outswapped processes.

- Hibernate (HIB)—Hibernating, or process has executed a hibernate request; resident processes.

- Hibernate (Outswapped) (HIBO)—Hibernating, or process has executed a hibernate request; outswapped processes.

- Suspended (SUSP)—Process has executed a suspend request; resident processes.

- Suspended (Outswapped) (SUSPO)—Process has executed a suspend request; outswapped processes.

- Free Page Wait (FPW)—Waiting for a free page of memory.

- Compute (COM)—Ready to use the processor; resident processes.

- Compute (Outswapped) (COMO)—Ready to use the processor; outswapped processes.

• Current Process (CUR)—Using the processor.

The data items can also be displayed as percentages of all processes.

Note that the Current Process is always the process running MONITOR, since MONITOR is running when each measurement is made.

For performance reasons, MONITOR does not synchronize the scanning of process state data structures with operating system use of those structures. It is therefore possible that MONITOR will display certain anomalous state indications.

EXAMPLE

```
$ MONITOR/INPUT/SUMMARY/NODISPLAY -
_$/BEGINNING=31-DEC-1988:13:00 -
_$/ENDING=31-DEC-1988:14:00 STATES/PERCENT/ALL
$ TYPE MONITOR.SUM
```

```
                        VAX/VMS Monitor Utility
                        PROCESS STATES (%)
                        on node SAMPLE        From: 31-DEC-1988 13:00:00
                        SUMMARY                 To: 31-DEC-1988 14:00:00
```

	CUR%	AVE%	MIN%	MAX%
Collided Page Wait	0.0	0.0	0.0	0.0
Mutex & Misc Resource Wait	0.0	0.0	0.0	0.0
Common Event Flag Wait	0.0	0.0	0.0	0.0
Page Fault Wait	4.3	1.4	0.0	4.3
Local Event Flag Wait	34.7	31.7	34.7	42.8
Local Evt Flg (Outswapped)	0.0	9.0	0.0	19.4
Hibernate	43.4	40.7	43.4	52.1
Hibernate (Outswapped)	0.0	4.3	0.0	15.4
Suspended	0.0	0.0	0.0	0.0
Suspended (Outswapped)	0.0	0.0	0.0	0.0
Free Page Wait	0.0	0.0	0.0	0.0
Compute	13.0	7.3	4.3	13.0
Compute (Outswapped)	0.0	0.8	0.0	3.2
Current Process	1.0	1.0	1.0	1.0

```
PLAYBACK                SUMMARIZING
```

The commands in this example generate and display a PROCESS STATES summary. Note that since use of the RETURN key is not permitted within a single MONITOR command following the MONITOR> prompt, the MONITOR command is entered at DCL level. The summary shows that, on the average, 14.1% of processes were swapped out for the summarized period. Note that the summary was requested for data covering only the hour between 1 P.M. and 2 P.M., although the input file could have contained data covering a longer period.

MONITOR SYSTEM

The MONITOR SYSTEM command initiates monitoring of the SYSTEM statistics class, which shows several of the most important items from other classes.

FORMAT   **MONITOR SYSTEM**

QUALIFIERS   */qualifier[,...]*
One or more qualifiers as described in the Qualifiers section.

CLASS-NAME   */ALL*
QUALIFIERS   Specifies that a table of all available statistics (current, average, minimum, and maximum) is to be included in the display and summary output. For summary output, this qualifier is the default for all classes; otherwise, it is the default for all classes except CLUSTER, MODES, PROCESSES, STATES, and SYSTEM.

/AVERAGE
Selects average statistics to be displayed in a bar graph for display and summary output.

/CURRENT
Selects current statistics to be displayed in a bar graph for display and summary output. The /CURRENT qualifier is the default for the CLUSTER, MODES, STATES, and SYSTEM classes.

/MAXIMUM
Selects maximum statistics to be displayed in a bar graph for display and summary output.

/MINIMUM
Selects minimum statistics to be displayed in a bar graph for display and summary output.

DESCRIPTION   Because the SYSTEM class collects the most significant performance statistics from other classes in a single display, it is particularly useful to system managers and other users seeking a general overview of system activity. The SYSTEM class includes the following data items:

Interrupt Stack
MP Synchronization
Kernel Mode
Executive Mode
Supervisor Mode
User Mode
Compatibility Mode
Idle Time
Process Count

Page Fault Rate
Page Read I/O Rate
Free List Size
Modified List Size
Direct I/O Rate
Buffered I/O Rate

The following two display formats are provided, depending on the class-name qualifier specified:

- A tabular style format for the /ALL qualifier

- A bar graph style format for the /AVERAGE, /CURRENT, /MAXIMUM, and /MINIMUM qualifiers

Note that the bar graph version of the SYSTEM class (shown in Example 2) contains the following data, which differs from the tabular version:

- All of the CPU processor modes except Idle Time are summed in the CPU Busy segment.

- In the Page Fault segment, the page read I/O rate is indicated by a vertical bar. The bar provides a visual estimate of the proportion of the total page fault rate that caused read I/O operations (the "hard" fault rate). The "hard" fault rate appears to the left of the bar.

- Four segments show the processes that are currently (since the last screen update) the top consumers of CPU, page faults, direct I/Os, and buffered I/Os. Note that the NULL process is never displayed, even if it is a top consumer.

Any process that MONITOR designates as a top user process must be swapped in at the beginning and ending of the display interval or at the beginning and ending of the entire period covered by a summary.

When the lower bar graph (top user) and the corresponding upper bar graph (overall system measure) are tracking the same statistic for the same interval (as in Example 2), it is reasonable to compare the two graphs. This will be the case in the following situation:

- SYSTEM is the only class being monitored (no other class names have been specified with the MONITOR command).

- The CURRENT statistic is specified.

- The /INTERVAL and /VIEWING_TIME values are equal.

Otherwise, you should exercise care in making such comparisons because the top user statistic is always CURRENT, while the overall system measure statistic may be CURRENT, AVERAGE, MAXIMUM, or MINIMUM.

Rates for top users are calculated based on the interval between two successive screen displays, while overall system rates are based on the collection interval. These two interval values can be different whenever one or more classes are being monitored with the SYSTEM class, or when /INTERVAL and /VIEWING_TIME values differ.

While other upper boundary figures for the SYSTEM class bar graphs are constants, the figures for Free List Size and Modified List Size are derived from the physical memory configuration and system parameters of individual systems. The upper boundary figure for the Free List is the number of pages

available after deducting the pages permanently allocated to VMS. This figure, sometimes referred to as *balance set memory,* is the number of pages that can be allocated to processes, the Free List, and the Modified List. The upper boundary figure for the Modified List is the value of the MPW_HILIMIT system parameter. Note that both upper boundary figures are calculated when the MONITOR request is initiated and do not change thereafter.

EXAMPLES

1   MONITOR> MONITOR SYSTEM/ALL

```
                              VAX/VMS Monitor Utility
                                 SYSTEM STATISTICS
                                  on node SAMPLE
                              31-DEC-1988 12:43:28
                                  CUR       AVE       MIN       MAX

            Interrupt Stack       0.33      0.33      0.33      0.33
            MP Synchronization    0.00      0.00      0.00      0.00
            Kernel Mode           0.16      0.16      0.16      0.16
            Executive Mode        0.00      0.00      0.00      0.00
            Supervisor Mode       0.00      0.00      0.00      0.00
            User Mode             0.50      0.49      0.50      0.50
            Compatibility Mode    0.00      0.00      0.00      0.00
            Idle Time            99.00     98.67     99.00     99.00
            Process Count        14.00     14.00     14.00     14.00
            Page Fault Rate       0.33      0.33      0.33      0.33
            Page Read I/O Rate    0.00      0.00      0.00      0.00
            Free List Size     4255.00   4255.00   4255.00   4255.00
            Modified List Size  105.00    105.00    105.00    105.00
            Direct I/O Rate       0.00      0.00      0.00      0.00
            Buffered I/O Rate     0.16      0.16      0.16      0.16
```

This example shows the tabular style format for the SYSTEM display.

2   MONITOR> MONITOR SYSTEM

```
Node: SAMPLE                VAX/VMS Monitor Utility      31-DEC-1988 12:38:48
Statistic: CURRENT              SYSTEM STATISTICS
                                                 Process States
           + CPU Busy (100)           -+      LEF:    7        LEFO:    0
           |************************|          HIB:   11        HIBO:    0
CPU   0 +---------------------------+ 100      COM:    4        COMO:    0
           |********                |          PFW:    0        Other:   1
           +---------------------------+        MWAIT:  0
           Cur Top: ASSEM_LIBRTL (34)                  Total: 23

           + Page Fault Rate (25)    -+        + Free List Size (4604)   -+
           |**|***                   |         |*********              | 12K
MEMORY 0 +---------------------------+ 100   0 +---------------------------+
           |**                       |         |*******               | 500
           +---------------------------+        + Modified List Size (146) +
           Cur Top: ASSEM_SYS (11)

           + Direct I/O Rate (15)    -+        + Buffered I/O Rate (0)   -+
           |******                   |         |                       |
I/O   0 +---------------------------+ 60    0 +---------------------------+ 150
           |****                     |         |                       |
           +---------------------------+        +---------------------------+
           Cur Top: ASSEM_SYS (10)             Cur Top: Cerb Servant (0)
```

This example shows the bar graph style format for the SYSTEM display.

SET DEFAULT

The SET DEFAULT command sets command qualifier and class-name parameter defaults for the MONITOR command. Each SET DEFAULT command sets only the qualifiers you specify, but replaces the entire set of class-name parameters.

FORMAT	**SET DEFAULT** *classname[,...]*

PARAMETERS	*classname[,...]* Specifies one or more class names.

QUALIFIERS	*/qualifier[,...]* One or more qualifiers as described in the Qualifiers section.

CLASS-NAME QUALIFIERS	*/qualifier[,...]* One or more class-name qualifiers (see Table MON-1).

DESCRIPTION	Command and class-name qualifiers are identical to those for the MONITOR class-name commands.

EXAMPLE

```
MONITOR> SET DEFAULT /INTERVAL=10 PAGE/AVERAGE+POOL/MAXIMUM /NODE=(LARRY,MOE,CURLEY)
```

The command in this example selects PAGE and POOL as the default classes for the MONITOR command and specifies an interval of 10 seconds for the statistics display. The command specifies that AVERAGE statistics be displayed for the PAGE class, and that MAXIMUM statistics be displayed for the POOL class. Finally, the command requests that data be collected on nodes LARRY, MOE, and CURLEY. After establishing these defaults, you can enter the MONITOR command without any qualifiers or parameters to display the specified results.

SHOW DEFAULT

The SHOW DEFAULT command displays the defaults established by the SET DEFAULT command.

FORMAT   **SHOW DEFAULT**

PARAMETERS   *None.*

QUALIFIERS   *None.*

DESCRIPTION   You can use the SHOW DEFAULT command to verify the defaults you have set with the SET DEFAULT command.

EXAMPLE

```
MONITOR> SHOW DEFAULT

/BEGINNING = current time          /INTERVAL     = 10
/ENDING    = indefinite            /VIEWING_TIME = 10

/FLUSH_INTERVAL = 300

/NOINPUT

/NORECORD

/DISPLAY   = SYS$OUTPUT:.;

/NOSUMMARY

/NOFILENAME

/NOCOMMENT

Classes:
 PAGE/AVERAGE POOL/MAXIMUM

Nodes:
 LARRY              MOE              CURLEY
```

The command in this example displays the defaults specified by the previous SET DEFAULT command.

MONITOR
EXAMPLES

LIVE DISPLAY
MONITORING
EXAMPLES

The following examples illustrate the live display monitoring mode of operation. Use this mode when you want to examine the activity of a running system, either on a routine basis, or as part of an installation checkout, tuning, or trouble-shooting exercise. No historical record of output is kept.

1   `$ MONITOR PROCESSES/TOPCPU`

In this example, the command displays a bar graph showing the eight processes that were the top consumers of CPU time during the period between displays. It also displays the amount of CPU time each of these processes used.

2   `$ MONITOR/DISPLAY=POOL.LOG POOL`

MONITOR display output can be routed to any supported terminal device or to a disk file. This command writes MONITOR's display of nonpaged pool statistics to the file POOL.LOG. This file could then be printed out on a hardcopy device.

3   `$ MY_CLASSES :== -`
`_$ "DECNET+FCP+IO+LOCK+MODES+PAGE+POOL+PROCESSES+STATES"`
`$ MONITOR/NODE=(CURLEY,LARRY)/INTERVAL=20/VIEWING_TIME=8 'MY_CLASSES'`

You may find it convenient to establish DCL symbols for frequently used combinations of class names, as in this example. The MONITOR command collects selected classes of data for nodes CURLEY and LARRY every 20 seconds. Every 8 seconds, the most recently collected data for one of the classes is displayed. The ordering of the classes for display is predetermined by MONITOR.

LIVE
RECORDING
EXAMPLE

The following example illustrates the live recording mode of operation. Use live recording when you need to capture MONITOR data for future use. Possible uses include the following:

- Installation checkout, tuning, trouble-shooting; that is, all the uses listed for live display monitoring. Choose recording over display when you want to capture more classes than you could reasonably watch at a terminal, when a terminal is not available, or when you need to gather data about the system but cannot devote your time to the terminal until later.

- Routine performance data gathering for long-term analysis. MONITOR data can be recorded on a routine basis and summarized to gather data about system resource utilization over long periods of time.

```
$ MONITOR/NODE=(LARRY,MOE)/NODISPLAY/RECORD MODES+STATES
```

> This command records data on the time spent in each of the processor modes
> and on the number of processes in each of the scheduler states for nodes
> LARRY and MOE. It does not display this information.

CONCURRENT DISPLAY AND RECORDING EXAMPLES

The following examples illustrate the concurrent display and recording mode
of operation. Use this mode when you want to both retain performance data
and watch as it is being collected.

The last two examples show how you can perform concurrent recording
and display using two separate processes; one process records and the other
displays. Because MONITOR allows shared read access to the recording file,
the display process can play back the recording file as it is being written. The
process in the second example performs recording. The process in the third
example plays back the file to obtain a summary.

1
```
$ MONITOR/RECORD FCP/AVERAGE,POOL/MINIMUM
```

> This command collects and records file system data and nonpaged dynamic
> pool data every three seconds. It also displays, in bar graph form, average
> FCP statistics and minimum POOL statistics. The display alternates between
> the two graphs every three seconds. Current statistics can be obtained in a
> subsequent playback request.

2
```
$ MONITOR/RECORD=SYS$MANAGER:ARCHIVE.DAT: -
_$ /INTERVAL=300/NODISPLAY ALL_CLASSES)
```

> The recording process uses this command to archive data for all classes once
> every three minutes. System managers may find it convenient to place a
> similar command in their system startup command file.

3
```
$ MONITOR/INPUT=SYS$MANAGER:ARCHIVE.DAT: -
_$ /NODISPLAY/SUMMARY/BEGINNING="-1" PAGE,IO)
```

> The display process issues this command to obtain a summary of page and
> I/O activity that occurred during the previous hour, perhaps as part of
> an investigation of a reported performance problem. Note that since the
> recording process executes an RMS flush operation every five minutes, up to
> five minutes of the most recently collected data will not be available to the
> display process. The time between flush operations can be specified explicitly
> with the /FLUSH_INTERVAL qualifier. Note also that the display process
> must have read access to the recording file.

MONITOR
MONITOR Examples

PLAYBACK
EXAMPLES

The following examples illustrate the playback mode of operation. Use playback of a recording file to obtain terminal output and summary reports of all or just a subset of collected data. Data can be made into a subset according to class or time segment. For example, if several classes of data have been collected for an entire day, you can examine or summarize the data on one or more of the classes during any time period in that day. You can also display or summarize data with a different interval than the one at which it was recorded. The actual amount of time between displays of screen images is controlled with the /VIEWING_TIME qualifier.

1
```
$ MONITOR/RECORD/INTERVAL=5 IO
   .
   .
   .
$ MONITOR/INPUT IO
```

The commands in this example produce system I/O statistics. The first command gathers and displays data every five seconds, beginning when the command is entered and ending when the user enters a CTRL/C. In addition, it records binary data in the default output file MONITOR.DAT. The second command plays back the I/O statistics display, using the data in MONITOR.DAT for input. The default viewing time for the playback is three seconds, but each screen display represents five seconds of monitored I/O statistics.

2
```
$ MONITOR/RECORD/NODISPLAY -
_$ /BEGINNING=08:00:00 -
_$ /ENDING=16:00:00 -
_$ /INTERVAL=120 POOL
   .
   .
   .
$ MONITOR/INPUT/DISPLAY=HOURLY.LOG -
_$ /INTERVAL=3600 POOL
```

The sequence of commands in this example illustrates data recording with a relatively small interval and data playback with a relatively large interval. This is useful for producing average, minimum, and maximum statistics that cover a wide range of time, but have greater precision than if they had been gathered using the larger interval.

The first command records data on space allocation in the nonpaged dynamic pool for the indicated eight-hour period, using an interval of two minutes. The second plays the data back with an interval of one hour, storing display output in the file HOURLY.LOG. This file can then be typed or printed to show the cumulative pool utilization at each hour throughout the eight-hour period.

3
```
$ MONITOR/INPUT/NODISPLAY/SUMMARY=DAILY.LOG POOL/AVERAGE
```

The command in this example uses the recording file created in the previous example to produce a one-page summary report file showing the average statistics for the indicated eight-hour period. The summary report has the same format as a screen display, which in this case is a bar graph.

REMOTE PLAYBACK EXAMPLE

You can collect MONITOR data from any system to which your system has a DECnet connection. You can then display the data live on your local system. To implement the procedure, follow these steps:

1   In the default DECNET directory on each remote system, create a file named MONITOR.COM, similar to the following:

```
$ !
$ !     * Enable MONITOR remote playback *
$ !
$ MONITOR /NODISPLAY/RECORD=SYS$NET ALL_CLASSES
```

2   On your local system, define a logical name, as follows, for the remote system from which you wish to collect data:

```
$ DEFINE remotenodename_mon "node::""task=monitor"""
```

You may want to define, in a login command procedure, a series of such logical names for the various systems you wish to access.

To display the remote MONITOR data as it is being collected, enter a command in the following form:

```
$ MONITOR/INPUT=remotenodename_mon classnames
```

You can also place MONITOR.COM files in directories other than the default DECNET directory and use access control strings or proxy accounts to invoke these command files remotely.

When you invoke MONITOR on your local system, a process is created on the remote system that executes the MONITOR.COM command file. The remote system therefore experiences some associated CPU and DECnet overhead. You can regulate the overhead in the MONITOR.COM file by using the /INTERVAL qualifier and the list of class names.

RERECORDING EXAMPLE

The following example illustrates the rerecording mode of operation.

Rerecording is a combination of playback and recording. It can be useful for data reduction of recording files. When you play back an existing recording file, all MONITOR options are available to you; thus, you can choose to record a subset of the recorded classes and a subset of the recorded time segment and even a larger interval value.

All these techniques produce a new, smaller recording file at the expense of some of the recorded data. A larger interval value reduces the volume of the collected data, so displays and summary output produced from the newer recorded file will be less precise. Note that average rate values are not affected in this case, but average level values will be less precise (since the sample size is reduced), as will maximum and minimum values.

```
$ SUBMIT MONREC.COM
```

MONREC.COM contains the following commands:

```
$ MONITOR/NODISPLAY/RECORD/INTERVAL=60 -
        /BEGINNING=8:00/ENDING=16:00  DECNET,LOCK

$ MONITOR/INPUT/NODISPLAY/RECORD DECNET
```

The first command runs in a batch job, recording DECnet–VAX and lock management data once per minute, between the hours of 8 A.M. and 4 P.M. The second command, which is issued after the first command completes, rerecords the data by creating a new version of the MONITOR.DAT file, containing only the DECnet–VAX data.

CLUSTER MULTIFILE SUMMARY EXAMPLE

This section provides an example of procedures that you, as cluster manager, might use to create multifile summaries in a cluster environment (cluster summaries). Adapt the command procedures to suit conditions at your site.

To facilitate generation of cluster summaries and to perform other automatic MONITOR functions, you can create and execute command files similar to those illustrated in the accompanying procedure (and supplied in SYS$EXAMPLES). Note that you must define the logical names SYS$MONITOR and MON$ARCHIVE in SYSTARTUP.COM before executing any of the command files.

To produce a cluster summary, create and execute command files similar to the ones that follow.

1 MONITOR.COM—The following procedure archives the recording file and summary file from the previous boot and initiates continuous recording for current boot. (Note that this procedure does not purge recording files.)

```
$ SET VERIFY
$ !
$ !   MONITOR.COM
$ !
$ !   This command file is to be placed in a cluster-accessible directory
$ !   called SYS$MONITOR and submitted at system startup time as a detached
$ !   process via SUBMON.COM.  For each node, MONITOR.COM creates, in
$ !   SYS$MONITOR, a MONITOR recording file that is updated throughout the
$ !   life of the boot. It also creates, in MON$ARCHIVE, a summary file from
$ !   the recording file of the previous boot, along with a copy of that
$ !   recording file.  Logical name definitions for both cluster-accessible
$ !   directories, SYS$MONITOR and MON$ARCHIVE, should be included in SYSTARTUP.COM.
$ !
$ SET DEF SYS$MONITOR
$ SET NOON
$ PURGE MONITOR.LOG/KEEP:2
$ !
$ !   Compute executing node name and recording and summary file names
$ !   (incorporating node name and date).
$ !
$ NODE = F$GETSYI("NODENAME")
$ SEP = ""
```

```
$ IF NODE .NES. "" THEN SEP = "_"
$ DAY = F$EXTRACT (0,2,F$TIME())
$ IF F$EXTRACT(0,1,DAY) .EQS. " " THEN DAY = F$EXTRACT(1,1,DAY)
$ MONTH = F$EXTRACT(3,3,F$TIME())
$ ARCHFILNAM = "MON$ARCHIVE:"+NODE+SEP+"MON"+DAY+MONTH
$ RECFIL = NODE+SEP+"MON.DAT"
$ SUMFIL = ARCHFILNAM+".SUM"
$ !
$ !
$ !  Check for existence of recording file from previous boot and skip
$ !  summary if not present.
$ !
$ OPEN/READ/ERROR=NORECFIL RECORDING 'RECFIL'
$ CLOSE RECORDING
$ !
$ !
$ !  Generate summary file from previous boot.
$ !
$ MONITOR /INPUT='RECFIL' /NODISPLAY /SUMMARY='SUMFIL' -
$ ALL_CLASSES+MODE/ALL+STATES/ALL+SCS/ITEM=ALL+SYSTEM/ALL+DISK/ITEM=ALL
$ !
$ !
$ !  Compute subject string and mail summary file to cluster manager.
$ !
$ !
$ A=""""
$ B=" MONITOR Summary "
$ SUB = A+NODE+B+F$TIME()+A
$ MAIL/SUBJECT='SUB' 'SUMFIL' CLUSTER_MANAGER
$ !
$ !
$ !  Archive recording file and delete it from SYS$MONITOR.
$ !
$ COPY 'RECFIL' 'ARCHFILNAM'.DAT
$ DELETE 'RECFIL';*
$ !
$ NORECFIL:
$ SET PROCESS/PRIORITY=15
$ !
$ !
$ !  Begin recording for this boot.  The specified /INTERVAL value is
$ !  adequate for long-term summaries; you may require a smaller value
$ !  to get reasonable "semi-live" playback summaries (at the expense
$ !  of more disk space for the recording file).
$ !
$ MONITOR /INTERVAL=300 /NODISPLAY /RECORD='RECFIL' ALL_CLASSES
$ !
$ !
$ !  End of MONITOR.COM
$ !
```

2 SUBMON.COM—The following procedure submits MONITOR.COM as a detached process from SYSTARTUP.COM to initiate continuous recording for the current boot:

```
$ SET VERIFY
$ !
$ !  SUBMON.COM
$ !
$ !  This command file is to be placed in a cluster-accessible directory
$ !  called SYS$MONITOR.  At system startup time, for each node, it is
$ !  executed by SYSTARTUP.COM, following logical name definitions for
$ !  the cluster-accessible directories SYS$MONITOR and MON$ARCHIVE.
$ !
$ !
$ !  Submit detached MONITOR process to do continuous recording.
$ !
$ !
$ RUN   SYS$SYSTEM:LOGINOUT.EXE -
        /UIC=[1,4]             -
        /INPUT=SYS$MONITOR:MONITOR.COM  -
        /OUTPUT=SYS$MONITOR:MONITOR.LOG -
        /ERROR=SYS$MONITOR:MONITOR.LOG  -
        /PROCESS_NAME="Monitor" -
        /WORKING_SET=100 -
        /MAXIMUM_WORKING_SET=100 -
        /EXTENT=512/NOSWap
$ !
$ !
$ !  End of SUBMON.COM
$ !
```

3 MONSUM.COM—The following procedure generates daily and prime-time cluster summaries:

```
$ SET VERIFY
$ !
$ !  MONSUM.COM
$ !
$ !  This command file is to be placed in a cluster-accessible directory
$ !  called SYS$MONITOR and executed at the convenience of the cluster
$ !  manager.  The file generates both 24-hour and "prime time" cluster
$ !  summaries and resubmits itself to run each day at midnight.
$ !
$ SET DEF SYS$MONITOR
$ SET NOON
$ !
$ !  Compute file specification for MONSUM.COM and resubmit the file.
$ !
$ FILE = F$ENVIRONMENT("PROCEDURE")
$ FILE = F$PARSE(FILE,,,"DEVICE")+F$PARSE(FILE,,,"DIRECTORY")+F$PARSE(FILE,,,"NAME")
$ SUBMIT 'FILE' /AFTER=TOMORROW /NOPRINT
$ !
$ !  Generate 24-hour cluster summary.
$ !
$ !
$ MONITOR/INPUT=(SYS$MONITOR:*MON*.DAT;*,MON$ARCHIVE:*MON*.DAT;*) -
  /NODISPLAY/SUMMARY=MONSUM.SUM -
  ALL_CLASSES+DISK/ITEM=ALL+SCS/ITEM=ALL-
  /BEGIN="YESTERDAY+0:0:0.00" /END="TODAY+0:0:0.00" /BY_NODE
$ !
$ !
```

```
$ !  Mail 24-hour summary file to cluster manager and delete the file from
$ !  SYS$MONITOR.
$ !
$ !
$ MAIL/SUBJECT="Daily Monitor Clusterwide Summary" MONSUM.SUM CLUSTER_MANAGER
$ DELETE MONSUM.SUM;*
$ !
$ !  Generate prime-time cluster summary.
$ !
$ !
$ MONITOR/INPUT=(SYS$MONITOR:*MON*.DAT;*,MON$ARCHIVE:*MON*.DAT;*) -
  /NODISPLAY/SUMMARY=MONSUM.SUM -
  ALL_CLASSES+DISK/ITEM=ALL+SCS/ITEM=ALL-
  /BEGIN="YESTERDAY+9:0:0.00" /END="YESTERDAY+18:0:0.00" /BY_NODE
$ !
$ !
$ !  Mail prime-time summary file to cluster manager and delete the file
$ !  from SYS$MONITOR.
$ !
$ !
$ MAIL/SUBJECT="Prime-Time Monitor Clusterwide Summary" MONSUM.SUM CLUSTER_MANAGER
$ DELETE MONSUM.SUM;*
$ !
$ !  End of MONSUM.COM
$ !
```

Note that MAIL commands in this procedure send files to user CLUSTER_MANAGER. Replace CLUSTER_MANAGER with the appropriate user name or logical name for your site.

Because summary data may be extensive, print out summary files.

A   Supplemental MONITOR Information—Record Formats

A.1   The MONITOR Recording File

Binary performance data is written into the MONITOR recording file when a MONITOR request indicates recording. A record is written to this file once per interval for each requested class. The record contains a predefined set of data for each of the requested performance classes.

The recording file is created when a MONITOR request is initiated, and is closed when the request terminates. The MONITOR recording file may be used as a source file to format and display the data on a terminal, to create a summary file, or to record a new recording file with different characteristics.

Note: **The record formats described in this section are subject to change without notice at any future VMS release.**

The 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. Following are the three categories of record types:

- Customer control record

- DIGITAL control record

- Class record

Customer control records may appear anywhere in the recording file. They are not generated by MONITOR and are ignored by MONITOR when it reads the file.

The first records in the MONITOR recording file, excluding customer control records, are DIGITAL control records. There are currently three types of DIGITAL control records at the beginning of the file: the file header record, the system information record, and the record RMS file name record. Node transition records are also control records, but can appear anywhere in the file.

Class records, which contain data on requested performance classes, follow the DIGITAL control records. The class record is generally written once per interval for each class being recorded. An exception to this rule occurs when several class records are required to contain data for a single class over a single interval. This case can occur for the PROCESSES class when there are too many processes to be accommodated by the maximum record size.

Unique numbers are assigned to each MONITOR record type. Record type numbers 0–127 are reserved for class records; numbers 128–191 are reserved for DIGITAL control records; numbers 192–255 are reserved for customer control records.

There are currently 26 record types generated MONITOR. Table A–1 lists the MONITOR record types and their numbers, with associated class types. (For an explanation of MONITOR class types, refer to Section A.4.1.)

Supplemental MONITOR Information—Record Formats
A.1 The MONITOR Recording File

Table A–1   MONITOR Record Types

	Type Number	Class Type
File Header	128	
System Information	129	
Node Transition	130	
RMS File Name	131	
PROCESSES Class	0	component
STATES Class	1	system
MODES Class	2	component
PAGE Class	3	system
IO Class	4	system
FCP Class	5	system
POOL Class	6	system
LOCK Class	7	system
DECNET Class	8	system
RESERVED	9	system
RESERVED	10	system
FILE_SYSTEM_CACHE Class	11	system
DISK Class	12	component
RESERVED	13	component
DLOCK Class	14	system
SCS Class	15	component
RESERVED	16	system
SYSTEM Class	17	system
RESERVED	18	system
CLUSTER Class	19	system
RMS Class	20	component
MSCP Class	21	system

A.2   Conventions

The following sections define the contents of each field within each record type. Record type and record size are given in decimal representation. References to system time indicate time values in system time format (64-bit format).

The field offset names listed are not defined within MONITOR. However, DIGITAL recommends that you define and use these offset names when you work with MONITOR output records.

Following is the suggested naming convention for the field offset names:

MNR_CCC$X_DDDDD

CCC is a record type or class mnemonic.

X is a one-letter code indicating the size of the data item, as follows:

B for byte
W for word
L for longword
Q for quadword
O for octaword
T for ASCII string

DDDDD is the name describing the data item.

In the following tables, the size of the data is shown in parentheses following the description of the field contents.

A.3   DIGITAL Control Records

There are four types of DIGITAL control records: the file header record, the system information record, the node transition record, and the RMS file record. Each file has one header record, which contains information applicable to all classes of performance data contained in the file. It must be the first record (except for customer control records) in the file.

There is one system information record per node per file. The record contains information about the system being monitored and follows the header record in the file.
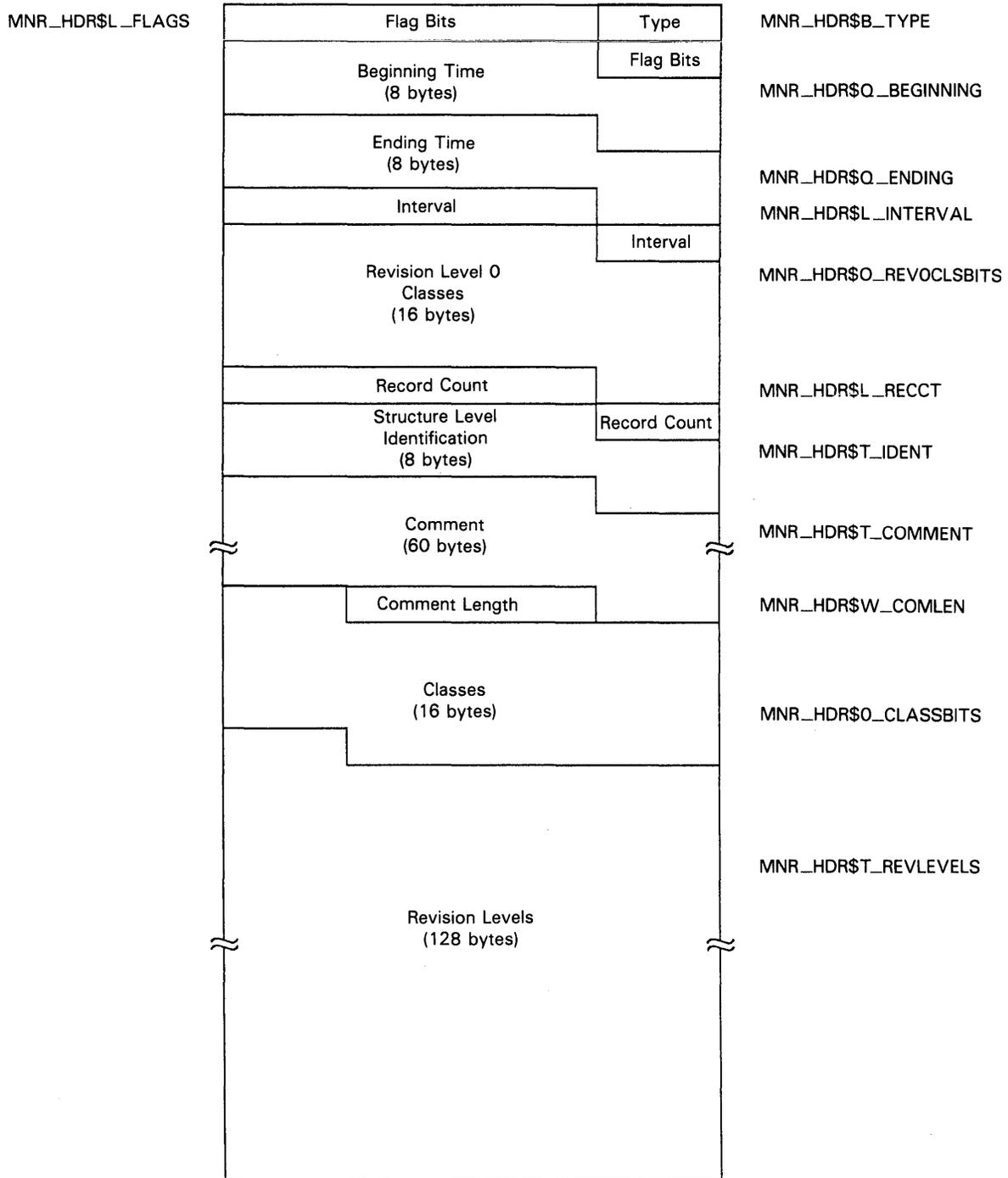
A.3.1   File Header Record

The file header record has a record type of 128 and a size of 259 bytes. Figure A–1 illustrates the format of the file header record; Table A–2 describes the fields in this record.

Supplemental MONITOR Information—Record Formats
A.3 DIGITAL Control Records

Figure A–1 File Header Record Format

ZK-966-82

Table A–2 Descriptions of File Header Record Fields

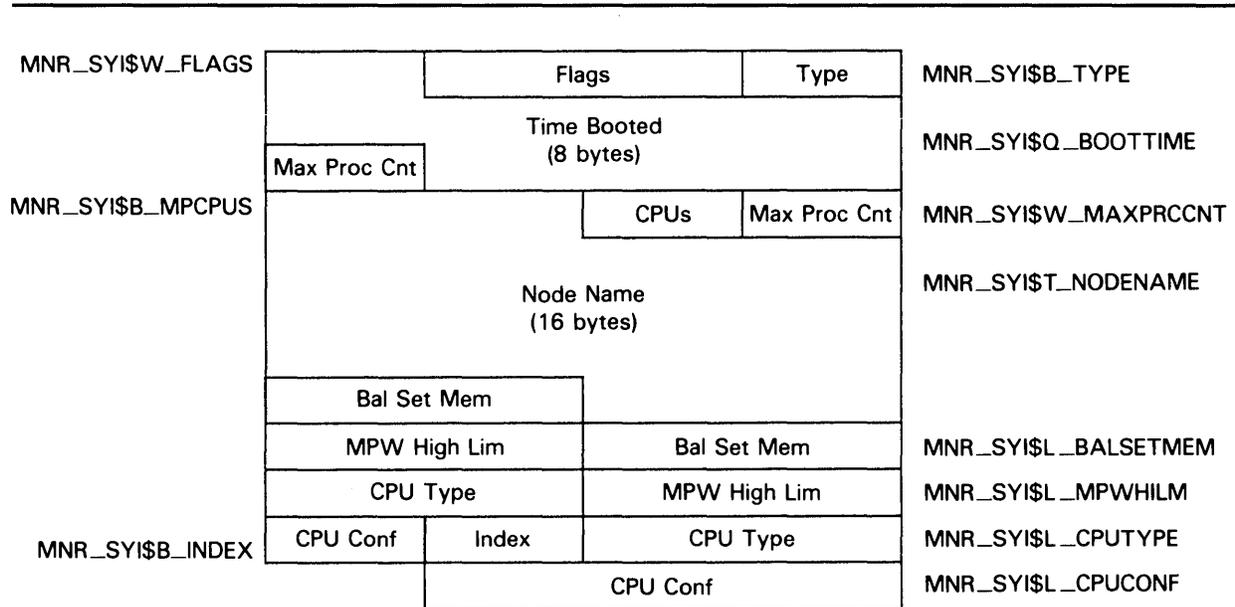| Field	Symbolic Offset	Contents
Type	MNR_HDR$B_TYPE	Record type identifier (1 byte).
Flags	MNR_HDR$L_FLAGS	32 flag bits; low-order bit = bit 0. All flags reserved to DIGITAL for future use (1 longword).
Beginning	MNR_HDR$Q_BEGINNING	System time of beginning of recording (1 quadword).
Ending	MNR_HDR$Q_ENDING	System time of end of recording (1 quadword).
Interval	MNR_HDR$L_INTERVAL	Interval in seconds between collections; this is the value specified by the user in the recording request. It is not necessarily equal to the exact interval value obtained by subtracting two consecutive time stamps for a given class (1 longword).
Revision Level 0	MNR_HDR$O_REV0CLSBITS	128-bit string representing all classes; a bit set to 1 indicates the presence in this file of a class which is at Revision Level 0 and whose type number corresponds to the bit number. Low-order bit = bit 0. (1 octaword). This field is provided for compatibility with Version 3.0 files.
Count	MNR_HDR$L_RECCT	Count of all records in the file (1 longword).
ID	MNR_HDR$T_IDENT	MONITOR Recording File Structure Level Identification (MON25050) (8 bytes).
Comment	MNR_HDR$T_COMMENT	Recording file description supplied by the user, including trailing blanks (60 bytes).
Comment Length	MNR_HDR$W_COMLEN	Actual length of recording file description string specified by the user (1 word).
Classes	MNR_HDR$O_CLASSBITS	128-bit string representing all classes; a bit set to 1 indicates the presence in this file of the class whose type number corresponds to the bit number. Low-order bit = bit 0 (1 octaword).
Revision Levels	MNR_HDR$T_REVLEVELS	128-byte string consisting of a 1-byte binary revision level number for each class. A class has a revision level of 0 initially. For each MONITOR release, if the record definition has changed, the revision level will be increased (not necessarily by 1).

A.3.2 System Information Record

The system information record has a record type of 129 and a size of 47 bytes. Figure A–2 illustrates the format of the system information record; Table A–3 describes the fields in this record.

Supplemental MONITOR Information—Record Formats

A.3 DIGITAL Control Records

Figure A–2 System Information Record Format

```
MNR_SYI$W_FLAGS  ┌─────────────┬───────────────────────┬─────────────┐
                 │             │         Flags         │    Type     │  MNR_SYI$B_TYPE
                 │             ├───────────────────────┴─────────────┤
                 │             │            Time Booted              │  MNR_SYI$Q_BOOTTIME
                 │ Max Proc Cnt│            (8 bytes)                 │
MNR_SYI$B_MPCPUS ├─────────────┴───────────┬───────────┬────────────┤
                 │                          │   CPUs    │Max Proc Cnt│  MNR_SYI$W_MAXPRCCNT
                 │                          └───────────┴────────────┤
                 │               Node Name                           │  MNR_SYI$T_NODENAME
                 │               (16 bytes)                          │
                 ├─────────────────────────┬─────────────────────────┤
                 │      Bal Set Mem        │                         │
                 ├─────────────────────────┼─────────────────────────┤
                 │      MPW High Lim       │      Bal Set Mem        │  MNR_SYI$L_BALSETMEM
                 ├─────────────────────────┼─────────────────────────┤
                 │        CPU Type         │      MPW High Lim       │  MNR_SYI$L_MPWHILM
                 ├───────────┬─────────────┼─────────────────────────┤
MNR_SYI$B_INDEX  │  CPU Conf │    Index    │        CPU Type         │  MNR_SYI$L_CPUTYPE
                 ├───────────┴─────────────┴─────────────────────────┤
                 │                  CPU Conf                          │  MNR_SYI$L_CPUCONF
                 └───────────────────────────────────────────────────┘
```

ZK-1984-84

Table A–3 Descriptions of System Information Record Fields

Field	Symbolic Offset	Contents
Type	MNR_SYI$B_TYPE	Type identifier (1 byte).
Flags	MNR_SYI$W_FLAGS	16 flag bits; low-order bit = bit 0. If bit 0 is set to 1, the node on which the data was collected is a member of a VAXcluster. All other flags reserved to DIGITAL for future use (1 word).
Time Booted	MNR_SYI$Q_BOOTTIME	System time at which system booted (1 quadword).
Max Process	MNR_SYI$W_MAXPRCCNT	MAXPROCESSCNT system parameter value (1 word).
CPUs	MNR_SYI$B_MPCPUS	Number of CPUs. (1 byte).
Node Name	MNR_SYI$T_NODENAME	Node name of node being monitored (counted ASCII string, 16 bytes).
Balance Set Memory	MNR_SYI$L_BALSETMEM	Number of process pages to which memory can be allocated (1 longword).
MPW High Limit	MNR_SYI$L_MPWHILM	MPW_HILIMIT system parameter value (1 longword).
CPU Type	MNR_SYI$L_CPUTYPE	CPU type code. Use $PRDEF macro for code values (1 longword).
Index	MNR_SYI$B_INDEX	Identifies the position of this node in several internal MONITOR data structures (1 byte).
CPU Config	MNR_SYI$L_CPUCONF	Bit mask defining the location of each CPU in a multiprocessor (1 longword).

A.3.3 Node Transition Record

The node transition record has a record type of 130 and a size of two bytes. Figure A-3 illustrates the format of the node transition record; Table A-4 describes the fields in this record.

Figure A-3 Node Transition Record Format

MNR—NTR$B—INDEX

Index (Byte)	Type (Byte)

MNR—NTR$B—TYPE

ZK-4532-85

Table A-4 Descriptions of Node Transition Record Fields

Field	Symbolic Offset	Contents
Type	MNR_NTR$B_TYPE	Record type identifier—indicates node removal operation (1 byte).
Index	MNR_NTR$B_INDEX	Identifies the position of this node in several internal MONITOR data structures (1 byte).

A.3.4 RMS File Record

The RMS file record has a record type of 131 and a variable size that depends on the number of RMS files and length of the file name string. Figure A-4 illustrates the format of the RMS file record; Table A-5 describes the fields in this record.

Figure A–4 RMS File Record Format
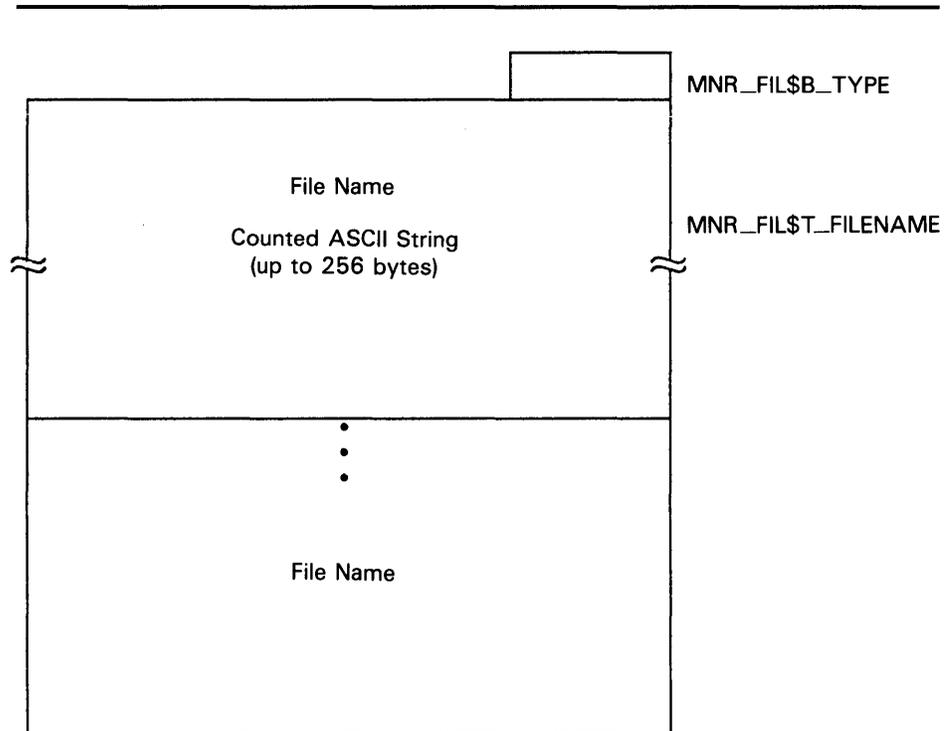
ZK-6107-HC

Table A–5 Descriptions of RMS File Record Fields

Field	Symbolic Offset	Contents
Type	MNR_FIL$B_TYPE	Record type identifier (1 byte).
Filename	MNR_FIL$T_FILENAME	A counted ASCII string that identifies the RMS file for MONITOR RMS requests. (up to 256 bytes).

A.4 Class Records

The MONITOR recording file contains one class record for each requested class for every collection interval, except for the PROCESSES class. (See Section A.4.2.13 for more information on the PROCESSES class records.) For example, if a MONITOR user requested to record five classes (excluding PROCESSES) for a duration of 100 collection intervals, the file would contain 500 class records. Class records occur in order of increasing type number within an interval. The first class record for a given interval follows the last class record for the previous interval.

A.4.1 Class Type Formats

There are two basic class types: system classes and component classes. A class record for a system class generally consists of counts for systemwide activities (such as page faults), whereas a class record for a component class normally contains a count for each element of a measured activity (such as I/O operations for each disk in the system).

Specifically, a class record for a system class consists of a class header followed by a data block. A class record for a component class has a class header followed by a class prefix and one data block per element.

Figure A–5 illustrates the format for class records.

Figure A–5 Class Record Format

SYSTEM CLASSES   COMPONENT CLASSES

Class Header
Data Block

Class Header
Class Prefix
Data Block 1
Data Block 2
• • •
Data Block n
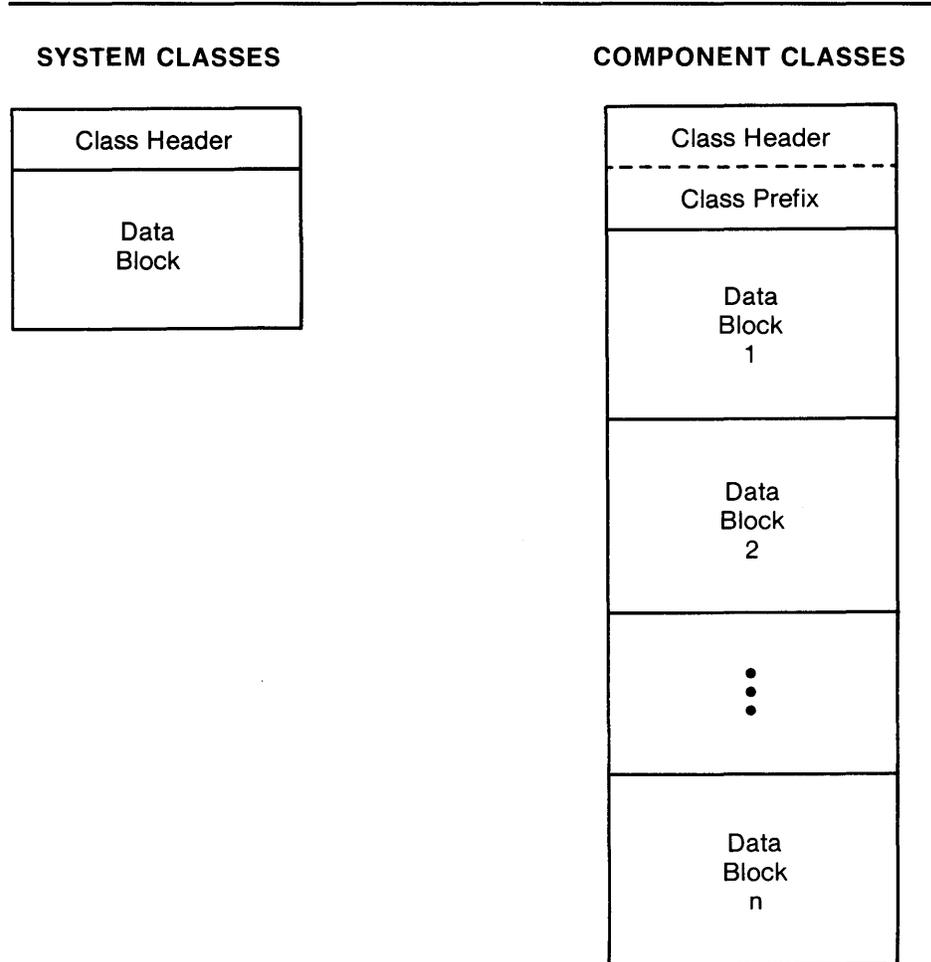
ZK-968-82

Supplemental MONITOR Information—Record Formats

A.4 Class Records

A.4.1.1 Class Header

The class header is the first part of every class record. Its format is independent of class. The class header is 13 bytes long.

Figure A–6 illustrates the format of the class header; Table A–6 describes the fields in the class header.

Figure A–6 Class Header Format
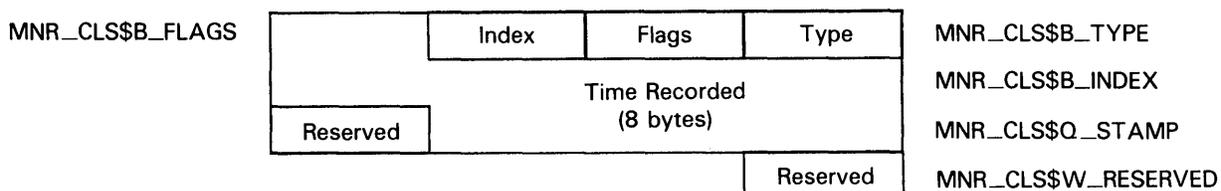
ZK-1983-84

Table A–6 Descriptions of Class Header Fields

Field	Symbolic Offset	Contents
Type	MNR_CLS$B_TYPE	Record type identifier (1 byte).
Flags	MNR_CLS$B_FLAGS	8 flag bits; low order bit = bit 0. If bit 0 is set to 1, the data for this interval continues in the next record. Can be set for the PROCESSES class only. All other flags reserved by DIGITAL for future use (I byte).
Index	MNR_CLS$B_INDEX	Identifies the position of this node in several internal MONITOR data structures (1 byte).
Time	MNR_CLS$Q_STAMP	System time at which this class record was recorded. The time value is nondecreasing across all class records in the file.
Reserved	MNR_CLS$W_RESERVED	Reserved for DIGITAL use (1 word).

A.4.1.2 Class Prefix (Component Classes Only)

The class prefix always follows the class header for component class records. It contains data describing the number of elements (processes for the PROCESSES class, disks for the DISK class) represented by the class records for the current collection interval. Unlike system class records, which have one data block per record, component classes have one data block per element.

One of the class prefix data items describes the number of elements (and therefore the number of data blocks) included in the class record. The other class prefix data item is used only for the PROCESSES class, and describes the number of processes included in the interval. The following discussion applies only to the PROCESSES class.

It is possible to monitor a number of processes so large that the required number of data blocks for one collection interval does not fit into a single maximum size record. In this case, the required number of PROCESSES class records is created to fully describe the processes.

All class headers in the set of PROCESSES class records for a given interval are identical, except for the setting of bit 0 in the MNR_CLS$W_FLAGS field. This bit is set to 1 for all records except the last, for which it is set to 0.

The class prefixes in the set of class records vary, as described in Table A-7. The contents of the MNR_CMP$L_ELTCT field depends on the number of data blocks contained in the record; the contents of the MNR_CMP$L_PCTINT field remain constant for each record in the set. All records in the set except the last contain as many data blocks as will fit into the maximum size record (32000 bytes). The last record in the set contains the remaining data blocks.

Figure A-7 illustrates the class prefix format; Table A-7 describes the fields in the class prefix. The class prefix is eight bytes long.

Figure A-7   Class Prefix Format

Elements in Record	MNR_CMP$L_ELTCT
Processes in Interval	MNR_CMP$L_PCTINT

ZK-970-82
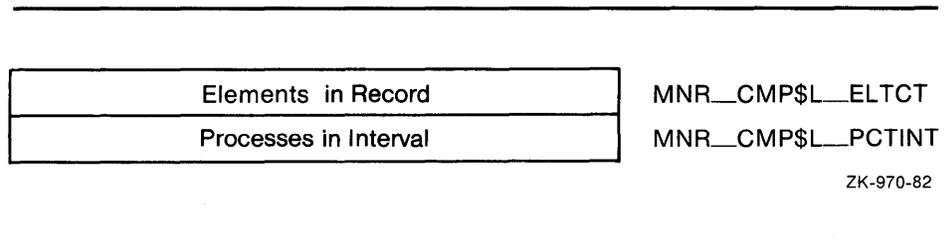
Table A-7   Descriptions of Class Prefix Fields

Field	Symbolic Offset	Contents
Elements In Record	MNR_CMP$L_ELTCT	Count of elements (data blocks) in this record (1 longword).
Processes In Interval	MNR_CMP$L_PCTINT	Count of processes (data blocks) for this interval (1 longword). This field is for the PROCESSES class only. For other component classes, this longword is reserved to DIGITAL for future use.

A.4.2   Class Data Blocks

The size and format of each data block and the number of blocks per record depend on the class. System classes have one data block per record. Component classes have one data block per element. The fields within each block are performance data items.

The following sections describe the data items within the data block for each class. Every data item falls into one of three categories. It is either a count, a level, or an informational item. A count is a numeric quantity that increases at each succeeding interval for the duration of a system boot. A level is a numeric quantity that may increase or decrease at each succeeding interval. An informational item represents data that, rather than being a unit of performance measurement (as are the first two types), is descriptive in nature.

In the tables that follow, item types are identified by the letters C (count), L (level), and I (informational). Item types are shown in parentheses, following the length of the field. Class records are listed alphabetically.
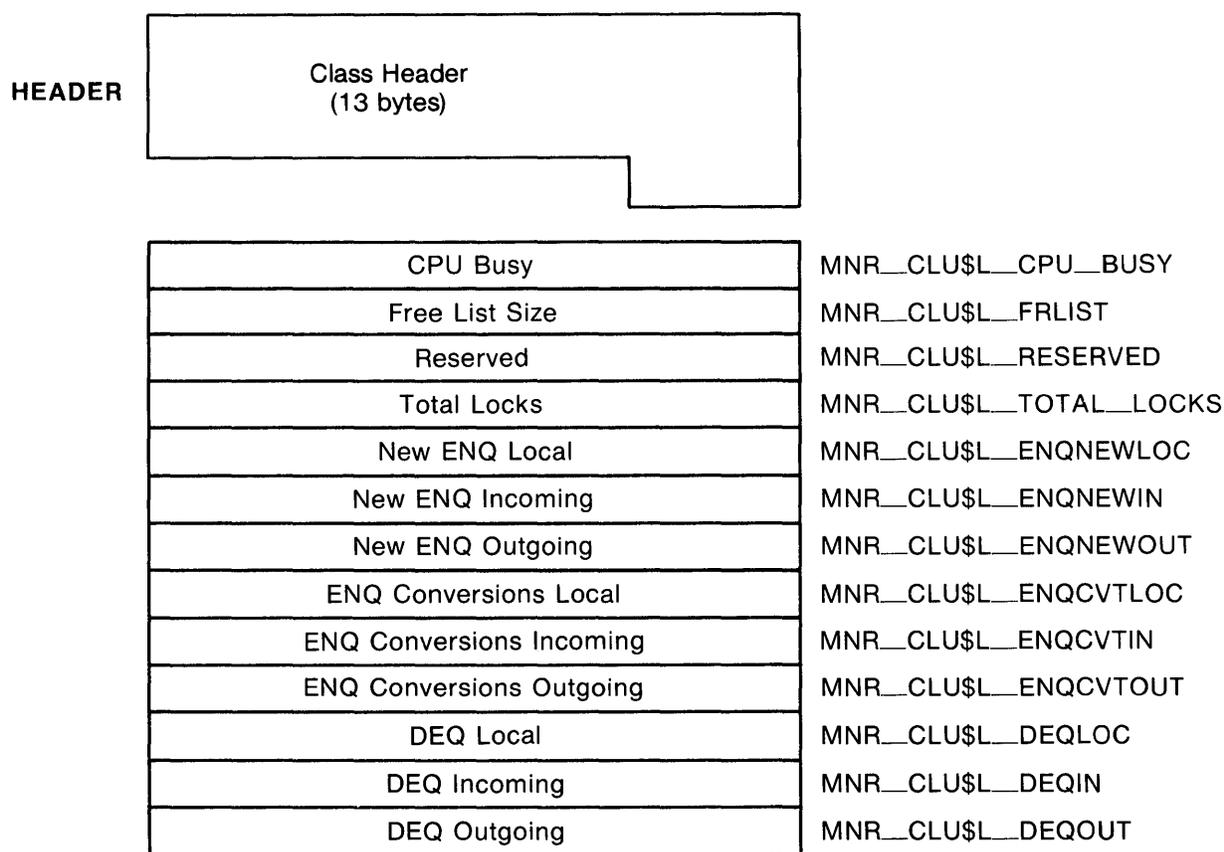
Supplemental MONITOR Information—Record Formats
A.4 Class Records

A.4.2.1 CLUSTER Class Record

The CLUSTER class record contains data describing clusterwide CPU, memory, and locking activity. The CLUSTER class record has a record type of 19 and a size of 65 bytes. Note that when the CLUSTER class is recorded, the DISK and MODES classes are also recorded, even if not explicitly requested.

Figure A–8 illustrates the format of the CLUSTER class record. Table A–8 describes the fields in the data block for the CLUSTER class record.

Figure A–8   CLUSTER Class Record Format

HEADER	Class Header (13 bytes)

CPU Busy	MNR__CLU$L__CPU__BUSY
Free List Size	MNR__CLU$L__FRLIST
Reserved	MNR__CLU$L__RESERVED
Total Locks	MNR__CLU$L__TOTAL__LOCKS
New ENQ Local	MNR__CLU$L__ENQNEWLOC
New ENQ Incoming	MNR__CLU$L__ENQNEWIN
New ENQ Outgoing	MNR__CLU$L__ENQNEWOUT
ENQ Conversions Local	MNR__CLU$L__ENQCVTLOC
ENQ Conversions Incoming	MNR__CLU$L__ENQCVTIN
ENQ Conversions Outgoing	MNR__CLU$L__ENQCVTOUT
DEQ Local	MNR__CLU$L__DEQLOC
DEQ Incoming	MNR__CLU$L__DEQIN
DEQ Outgoing	MNR__CLU$L__DEQOUT

ZK-4531-85

Table A–8 Descriptions of CLUSTER Class Record Fields

Field	Symbolic Offset	Contents
CPU Busy	MNR_CLU$L_CPU_BUSY	Count of clock ticks (10-millisecond units) spent in all CPU modes since system was booted (longword,C)
Free List Size	MNR_CLU$L_FRLIST	Number of pages currently on the free list (longword,L)
Reserved	MNR_CLU$L_RESERVED	Reserved to DIGITAL
Total Locks	MNR_CLU$L_TOTAL_LOCKS	Total of all incoming, outgoing, and local ENQs, DEQs, and conversions (longword,C)
New Enq Local	MNR_CLU$L_ENQNEWLOC	Count of new lock requests which originate and are performed on the system (local) (longword,C)
New Enq Incoming	MNR_CLU$L_ENQNEWIN	Count of new lock requests which originate on other systems and are performed on this system (incoming) (longword,C)
New Enq Outgoing	MNR_CLU$L_ENQNEWOUT	Count of new lock requests which originate on this system and are performed on other systems (outgoing) (longword,C)
Enq Conversions Local	MNR_CLU$L_ENQCVTLOC	Count of lock conversion requests (local) (longword,C)
Enq Conversions Incoming	MNR_CLU$L_ENQCVTIN	Count of lock conversion requests (incoming) (longword,C)
Enq Conversions Outgoing	MNR_CLU$L_ENQCVTOUT	Count of lock conversion requests (outgoing) (longword,C)
Deq Local	MNR_CLU$L_DEQLOC	Count of unlock requests (local) (longword,C)
Deq Incoming	MNR_CLU$L_DEQIN	Count of unlock requests (incoming) (longword,C)
Deq Outgoing	MNR_CLU$L_DEQOUT	Count of unlock requests (outgoing) (longword,C)

A.4.2.2 DECNET Class Record

The DECNET class record contains data describing the operation of the DECnet-VAX subsystem. The DECNET class record has a record type of 8 and a size of 37 bytes.

Figure A–9 illustrates the format of the DECNET class record. Table A–9 describes the fields in the data block for the DECNET class record.

Figure A–9   DECNET Class Record Format

ZK-979-82
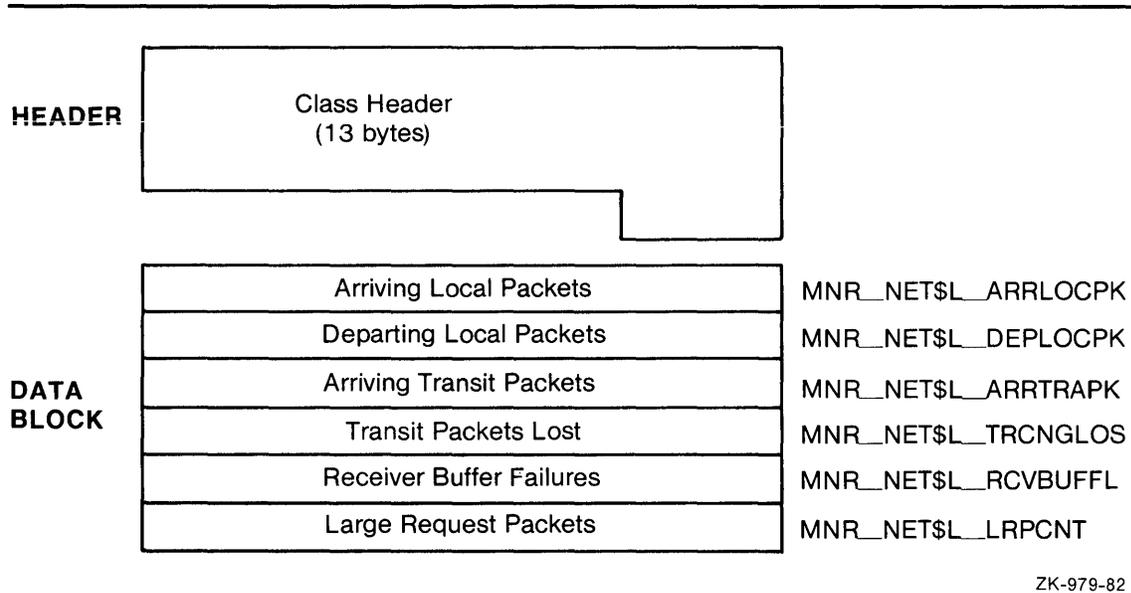
Table A–9   Descriptions of DECNET Class Record Fields

Field	Symbolic Offset	Contents
Arriving Local Packets	MNR_NET$L_ARRLOCPK	Count of arriving local packets (longword,C)
Departing Local Packets	MNR_NET$L_DEPLOCPK	Count of departing local packets (longword,C)
Arriving Transit Packets	MNR_NET$L_ARRTRAPK	Count of arriving transit packets (longword,C)
Transit Packets Lost	MNR_NET$L_TRCNGLOS	Count of packets lost because of transit congestion (longword,C)
Receiver Buffer Failures	MNR_NET$L_RCVBUFFL	Count of receiver buffer failures (longword,C)
Large Request Packets	MNR_NET$L_LRPCNT	Number of preallocated large request packets left in nonpaged pool (longword,L)

A.4.2.3   DISK Class Record

The DISK class record contains data describing all disk devices in the system. The DISK class record has a record type of 12; its size depends on the number of disks being monitored. The size, in bytes, is calculated by adding the size of the class header, the class prefix, and the data blocks contained in the record. This is shown in the following formula:

```
13 + 8 + (36 * the value of MNR_CMP$L_ELTCT)
```

Figure A–10 illustrates the format of the DISK class record. Table A–10 describes the fields in the data block for the DISK class record.

Figure A–10 DISK Class Record Format
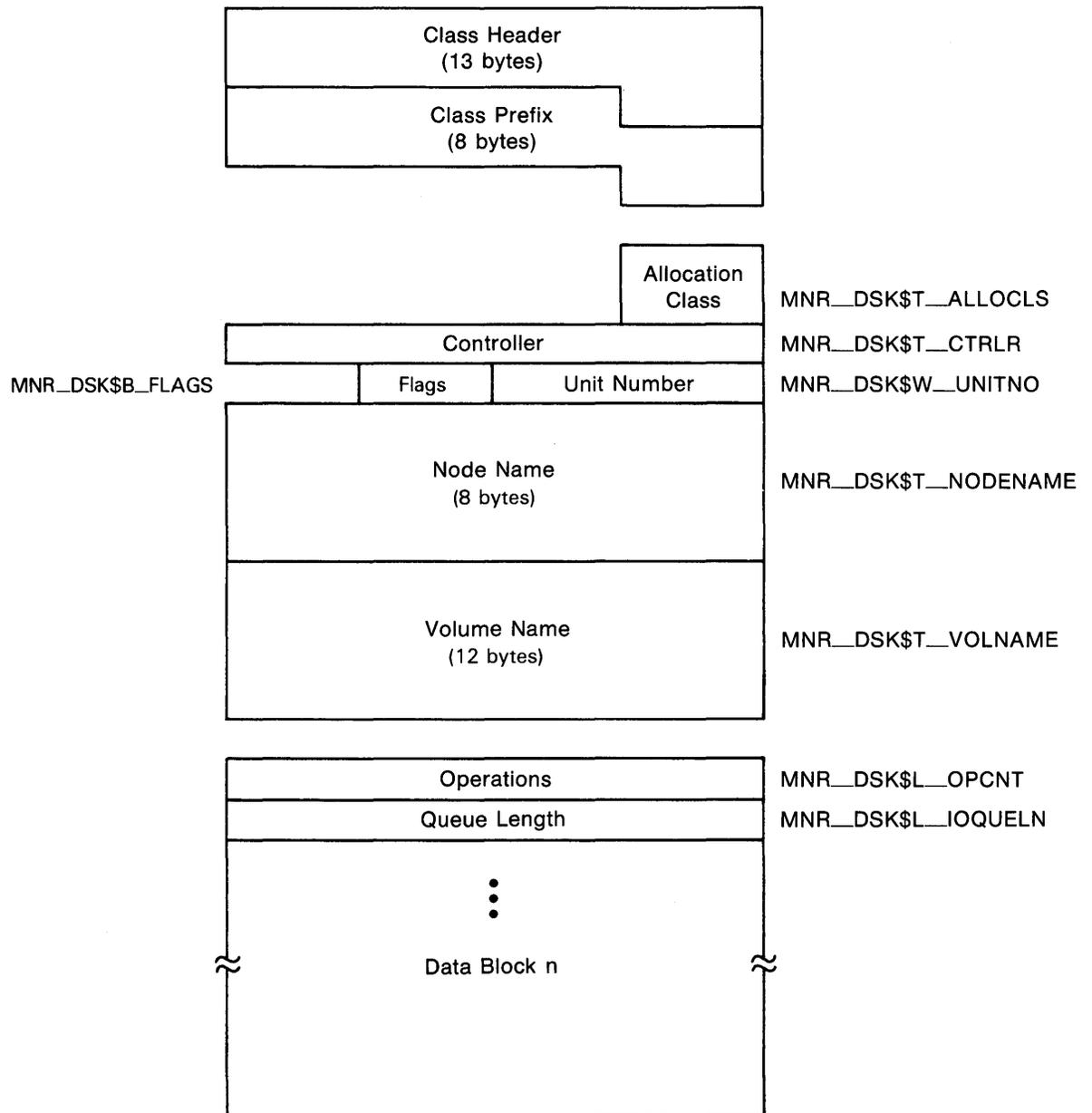
ZK-1987-84

Table A-10   Descriptions of DISK Class Record Fields

Field	Symbolic Offset	Contents
Allocation Class	MNR_DSK$B_ALLOCLS	Allocation class number (byte,I)
Controller	MNR_DSK$T_CTRLR	Name of device controller (counted ASCII string) (4 bytes,I)
Unit Number	MNR_DSK$W_UNITNO	Unit number (word,I)
Flags	MNR_DSK$B_FLAGS	8 flag bits; if the low bit is set, the device is served by the MSCP server (byte,I)
Node Name	MNR_DSK$T_NODENAME	Name of cluster node where device resides (counted ASCII string) (8 bytes,I)
Volume Name	MNR_DSK$T_VOLNAME	Volume name of disk (ASCII) (12 bytes,I)
Operations	MNR_DSK$L_OPCNT	Count of I/O operations (longword,C)
Queue Length	MNR_DSK$L_IOQUELN	Sum of I/O request queue samples (longword,C)

A.4.2.4   DLOCK Class Record

The DLOCK class record contains data describing the operation of the Distributed Lock Management Facility. The DLOCK class record has a record type of 14 and a size of 73 bytes.

Figure A-11 illustrates the format of the DLOCK class record. Table A-11 describes the fields in the data block for the DLOCK class record.

Figure A–11   DLOCK Class Record Format

HEADER	Class Header (15 bytes)	

DATA BLOCK	New Locks — Local	MNR_DLO$L_ENQNEWLOC
	New Locks — Incoming	MNR_DLO$L_ENQNEWIN
	New Locks — Outgoing	MNR_DLO$L_ENQNEWOUT
	Lock Conversions — Local	MNR_DLO$L_ENQCVTLOC
	Lock Conversions — Incoming	MNR_DLO$L_ENQCVTIN
	Lock Conversions — Outgoing	MNR_DLO$L_ENQCVTOUT
	Unlocks — Local	MNR_DLO$L_DEQLOC
	Unlocks — Incoming	MNR_DLO$L_DEQIN
	Unlocks — Outgoing	MNR_DLO$L_DEQOUT
	Blocking ASTs — Local	MNR_DLO$L_BLKLOC
	Blocking ASTs — Incoming	MNR_DLO$L_BLKIN
	Blocking ASTs — Outgoing	MNR_DLO$L_BLKOUT
	Functions Incoming	MNR_DLO$L_DIRIN
	Functions Outgoing	MNR_DLO$L_DIROUT
	Deadlock Message Rate	MNR_DLO$L_DLCKMSG

ZK-1349-83

Supplemental MONITOR Information—Record Formats
A.4 Class Records

Table A–11   Descriptions of DLOCK Class Record Fields

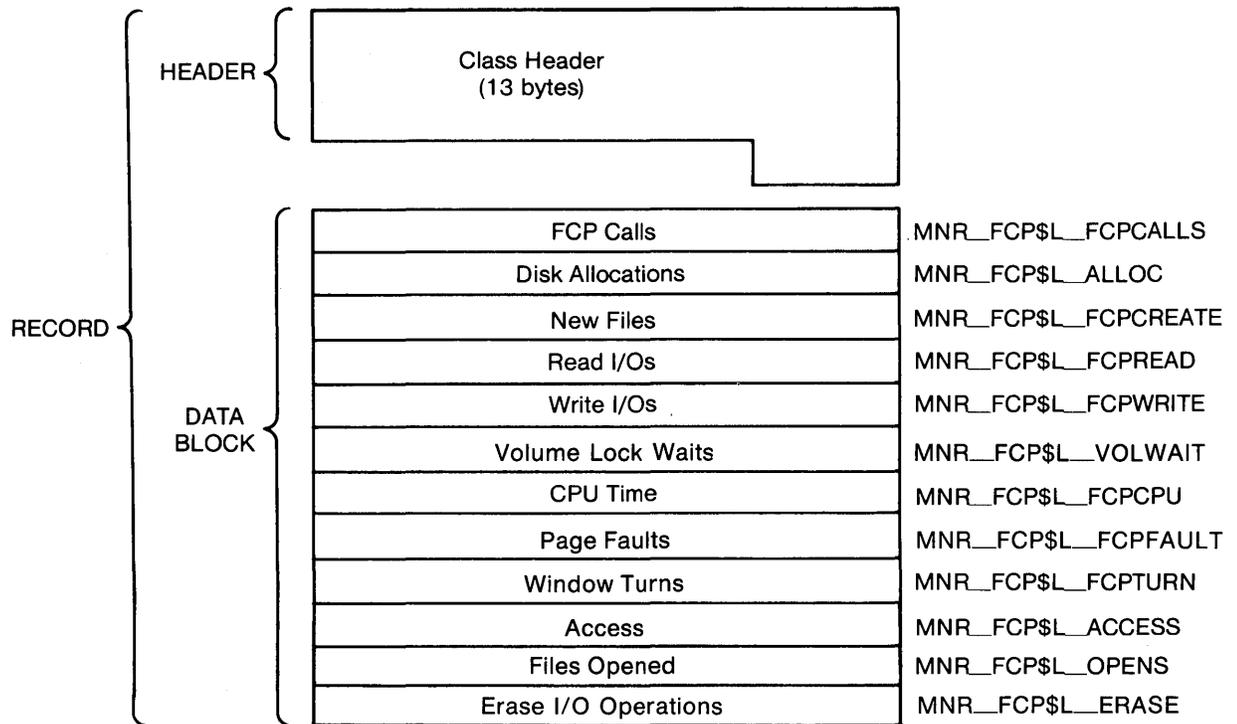| Field	Symbolic Offset	Contents
New Locks —Local	MNR_DLO$L_ENQNEWLOC	Count of new lock requests which originate and are performed on this system (local) (longword,C)
New Locks —Incoming	MNR_DLO$L_ENQNEWIN	Count of new lock requests originating on another system and performed on this system (incoming) (longword,C)
New Locks —Outgoing	MNR_DLO$L_ENQNEWOUT	Count of new lock requests originating on this system and performed on another system (outgoing) (longword,C)
Lock Conversions —Local	MNR_DLO$L_ENQCVTLOC	Count of lock conversion requests (local) (longword,C)
Lock Conversions —Incoming	MNR_DLO$L_ENQCVTIN	Count of lock conversion requests (incoming) (longword,C)
Lock Conversions —Outgoing	MNR_DLO$L_ENQCVTOUT	Count of lock conversion requests (outgoing) (longword,C)
Unlocks—Local	MNR_DLO$L_DEQLOC	Count of unlock requests (local) (longword,C)
Unlocks—Incoming	MNR_DLO$L_DEQIN	Count of unlock requests (incoming) (longword,C)
Unlocks—Outgoing	MNR_DLO$L_DEQOUT	Count of unlock requests (outgoing) (longword,C)
Blocking ASTs —Local	MNR_DLO$L_BLKLOC	Count of lock manager blocking ASTs (local) (longword,C)
Blocking ASTs —Incoming	MNR_DLO$L_BLKIN	Count of lock manager blocking ASTs (incoming) (longword,C)
Blocking ASTs —Outgoing	MNR_DLO$L_BLKOUT	Count of lock manager blocking ASTs (outgoing) (longword,C)
Directory Functions —Incoming	MNR_DLO$L_DIRIN	Count of directory functions (incoming) (longword,C)
Directory Functions —Outgoing	MNR_DLO$L_DIROUT	Count of directory functions (outgoing) (longword,C)
Deadlock Message Rate	MNR_DLO$L_DLCKMSG	Count of incoming and outgoing lock manager messages required for deadlock detection (longword,C)

A.4.2.5   FCP Class Record

The FCP class record contains data describing the operation of the file system ACPs. The FCP class record has a record type of 5 and a size of 61 bytes.

Figure A–12 illustrates the format of the FCP class record. Table A–12 describes the fields in the data block for the FCP class record.

Figure A–12   FCP Class Record Format


```
        ┌  ┌ ┌─────────────────────────────────────┐
        │  │ │        Class Header                 │
  HEADER┤  │ │        (13 bytes)                   │
        │  └ └────────────────────────────────┐    │
        │                                      └────┘
        │     ┌ ┌──────────────────────────┐
        │     │ │       FCP Calls          │  MNR_FCP$L_FCPCALLS
        │     │ ├──────────────────────────┤
        │     │ │     Disk Allocations     │  MNR_FCP$L_ALLOC
        │     │ ├──────────────────────────┤
        │     │ │       New Files          │  MNR_FCP$L_FCPCREATE
        │     │ ├──────────────────────────┤
  RECORD┤     │ │       Read I/Os          │  MNR_FCP$L_FCPREAD
        │     │ ├──────────────────────────┤
        │ DATA│ │       Write I/Os         │  MNR_FCP$L_FCPWRITE
        │BLOCK┤ ├──────────────────────────┤
        │     │ │    Volume Lock Waits     │  MNR_FCP$L_VOLWAIT
        │     │ ├──────────────────────────┤
        │     │ │       CPU Time           │  MNR_FCP$L_FCPCPU
        │     │ ├──────────────────────────┤
        │     │ │      Page Faults         │  MNR_FCP$L_FCPFAULT
        │     │ ├──────────────────────────┤
        │     │ │     Window Turns         │  MNR_FCP$L_FCPTURN
        │     │ ├──────────────────────────┤
        │     │ │        Access            │  MNR_FCP$L_ACCESS
        │     │ ├──────────────────────────┤
        │     │ │      Files Opened        │  MNR_FCP$L_OPENS
        │     │ ├──────────────────────────┤
        └     └ │   Erase I/O Operations   │  MNR_FCP$L_ERASE
                └──────────────────────────┘
```

ZK-976-82

Table A–12   Descriptions of FCP Class Record Fields

Field	Symbolic Offset	Contents
FCP Calls	MNR_FCP$L_FCPCALLS	Count of QIO requests received by the file system (longword,C)
Disk Allocations	MNR_FCP$L_ALLOC	Count of QIO requests that caused allocation of disk space (longword,C)
New Files	MNR_FCP$L_FCPCREATE	Count of new files created (longword,C)
Read I/Os	MNR_FCP$L_FCPREAD	Count of read I/O operations from the disk by the file system (longword,C)
Write I/Os	MNR_FCP$L_FCPWRITE	Count of write I/O operations to disk by the file system (longword,C)
Volume Lock Waits	MNR_FCP$L_VOLWAIT	Number of times a wait state was entered by the XQP due to volume lock contention (longword,C)
CPU Time	MNR_FCP$L_FCPCPU	Count of clock ticks (10-millisecond units) of CPU time used by the file system (longword,C)
FCP Page Faults	MNR_FCP$L_FCPFAULT	Count of page faults for the file system (longword,C)
Window Turns	MNR_FCP$L_FCPTURN	Count of file-map window misses (longword,C)

Supplemental MONITOR Information—Record Formats
A.4 Class Records

Table A–12 (Cont.)   Descriptions of FCP Class Record Fields

Field	Symbolic Offset	Contents
Access	MNR_FCP$L_ACCESS	Count of file name lookup operations in file directories (longword,C)
Files Opened	MNR_FCP$L_OPENS	Count of files opened (longword,C)
Erase I/O Operations	MNR_FCP$L_ERASE	Count of erase I/O operations issued (longword,C)

A.4.2.6   FILE_SYSTEM_CACHE Class Record

The FILE_SYSTEM_CACHE class record contains data describing the operation of the caches for the file system ACPs and XQPs. The FILE_SYSTEM_CACHE class record has a record type of 11 and a size of 69 bytes.

Figure A–13 illustrates the format of the FILE_SYSTEM_CACHE class record. Table A–13 describes the fields in the data block for the FILE_SYSTEM_CACHE class record.

Figure A–13 FILE_SYSTEM_CACHE Class Record Format

Class Header (13 bytes)

Directory FCB Cache Hits	MNR__FIL$L__DIRFCB__HIT
Directory FCB Cache Attempts	MNR__FIL$L__DIRFCB__TRIES
Directory Data Cache Hits	MNR__FIL$L__DIRDATA__HIT
Directory Data Cache Attempts	MNR__FIL$L__DIRDATA__TRIES
File Header Cache Hits	MNR__FIL$L__FILHDR__HIT
File Header Cache Attempts	MNR__FIL$L__FILHDR__TRIES
File ID Cache Hits	MNR__FIL$L__FIDHIT
File ID Cache Attempts	MNR__FIL$L__FID__TRIES
Extent Cache Hits	MNR__FIL$L__EXTHIT
Extent Cache Attempts	MNR__FIL$L__EXT__TRIES
Quota Cache Hits	MNR__FIL$L__QUOHIT
Quota Cache Attempts	MNR__FIL$L__QUO__TRIES
Storage Bitmap Cache Hits	MNR__FIL$L__STORAGMAP__HIT
Storage Bitmap Cache Attempts	MNR__FIL$L__STORAGMAP__TRIES

ZK-1985-84

Supplemental MONITOR Information—Record Formats
A.4 Class Records

Table A-13 Descriptions of FILE_SYSTEM_CACHE Class Record Fields

Field	Symbolic Offset	Contents
Directory FCB Cache Hits	MNR_FIL$L_DIRFCB_HIT	Count of hits on Directory FCB cache (longword,C)
Directory FCB Cache Attempts	MNR_FIL$L_DIRFCB_TRIES	Count of attempts on Directory FCB cache (longword,C)
Directory Data Cache Hits	MNR_FIL$L_DIRDATA_HIT	Count of hits on Directory Data cache (longword,C)
Directory Data Cache Attempts	MNR_FIL$L_DIRDATA_TRIES	Count of attempts on Directory Data cache (longword,C)
File Header Cache Hits	MNR_FIL$L_FILHDR_HIT	Count of hits on File Header cache (longword,C)
File Header Cache Attempts	MNR_FIL$L_FILHDR_TRIES	Count of attempts on File Header cache (longword,C)
File ID Cache Hits	MNR_FIL$L_FIDHIT	Count of hits on File ID cache (longword,C)
File ID Cache Attempts	MNR_FIL$L_FID_TRIES	Count of attemps on File ID cache (longword,C)
Extent Cache Hits	MNR_FIL$L_EXTHIT	Count of hits on Extent cache (longword,C)
Extent Cache Attempts	MNR_FIL$L_EXT_TRIES	Count of attempts on Extent cache (longword,C)
Quota Cache Hits	MNR_FIL$L_QUOHIT	Count of hits on Quota Cache (longword,C)
Quota Cache Attempts	MNR_FIL$L_QUO_TRIES	Count of attempts on Quota cache (longword,C)
Storage Bitmap Cache Hits	MNR_FIL$L_STORAGMAP_HIT	Count of hits on Storage Bitmap cache (longword,C)
Storage Bitmap Cache Attempts	MNR_FIL$L_STORAGMAP_TRIES	Count of attempts on Storage Bitmap cache (longword,C)

A.4.2.7   I/O Class Record

The I/O class record contains data describing the operation of the I/O subsystem. The I/O class record has a record type of 4 and a size of 69 bytes.

Figure A-14 illustrates the format of the I/O class record. Table A-14 describes the fields in the data block for the I/O class record.

Figure A–14 I/O Class Record Format

HEADER	Class Header (13 bytes)

	Direct I/Os
	Buffered I/Os
	Mailbox Writes
	Split Transfers
	Logical Name Translations
	Files Opened
DATA BLOCK	Page Faults
	Page Reads
	Page Read I/Os
	Page Writes
	Page Write I/Os
	Inswaps
	Free Page Count
	Modified Page Count

MNR_IO$L_DIRIO
MNR_IO$L_BUFIO
MNR_IO$L_MBWRITES
MNR_IO$L_SPLTRANS
MNR_IO$L_LOGNAM
MNR_IO$L_OPENS
MNR_IO$L_FAULTS
MNR_IO$L_PREADS
MNR_IO$L_PREADIO
MNR_IO$L_PWRITES
MNR_IO$L_PWRITIO
MNR_IO$L_ISWPCNT
MNR_IO$L_FREECNT
MNR_IO$L_MFYCNT

ZK-975-82

Table A–14 Descriptions of I/O Class Record Fields

Field	Symbolic Offset	Contents
Direct I/Os	MNR_IO$L_DIRIO	Count of direct I/O operations (longword,C)
Buffered I/Os	MNR_IO$L_BUFIO	Count of buffered I/O operations (longword,C)
Mailbox Writes	MNR_IO$L_MBWRITES	Count of write-to-mailbox requests (longword,C)
Split Transfers	MNR_IO$L_SPLTTRNS	Count of split transfers (longword,C)
Logical Name Translations	MNR_IO$L_LOGNAM	Count of logical name translations (longword,C)
Files Opened	MNR_IO$L_OPENS	Count of files opened (longword,C)
Page Faults	MNR_IO$L_FAULTS	Count of page faults for all working sets (longword,C)
Page Reads	MNR_IO$L_PREADS	Count of pages read from disk as a result of page faults (longword,C)
Page Read I/Os	MNR_IO$L_PREADIO	Count of read I/O operations from disk as a result of page faults (longword,C)
Page Writes	MNR_IO$L_PWRITES	Count of pages written to the page file (longword,C)

Table A-14 (Cont.) Descriptions of I/O Class Record Fields

Field	Symbolic Offset	Contents
Page Write I/Os	MNR_IO$L_PWRITIO	Count of write I/O operations to the page file (longword,C)
Inswaps	MNR_IO$L_ISWPCNT	Count of working sets read into memory from the swap file (longword,C)
Free Page Count	MNR_IO$L_FREECNT	Number of pages currently on free page list (longword,L)
Modified Page Count	MNR_IO$L_MFYCNT	Number of pages currently on modified page list (longword,L)

A.4.2.8 LOCK Class Record

The LOCK class record contains data describing the operation of the lock management subsystem. The LOCK class record has a record type of 7 and a size of 53 bytes.

Figure A-15 illustrates the format of the LOCK class record. Table A-15 describes the fields in the data block for the LOCK class record.

Figure A-15 LOCK Class Record Format
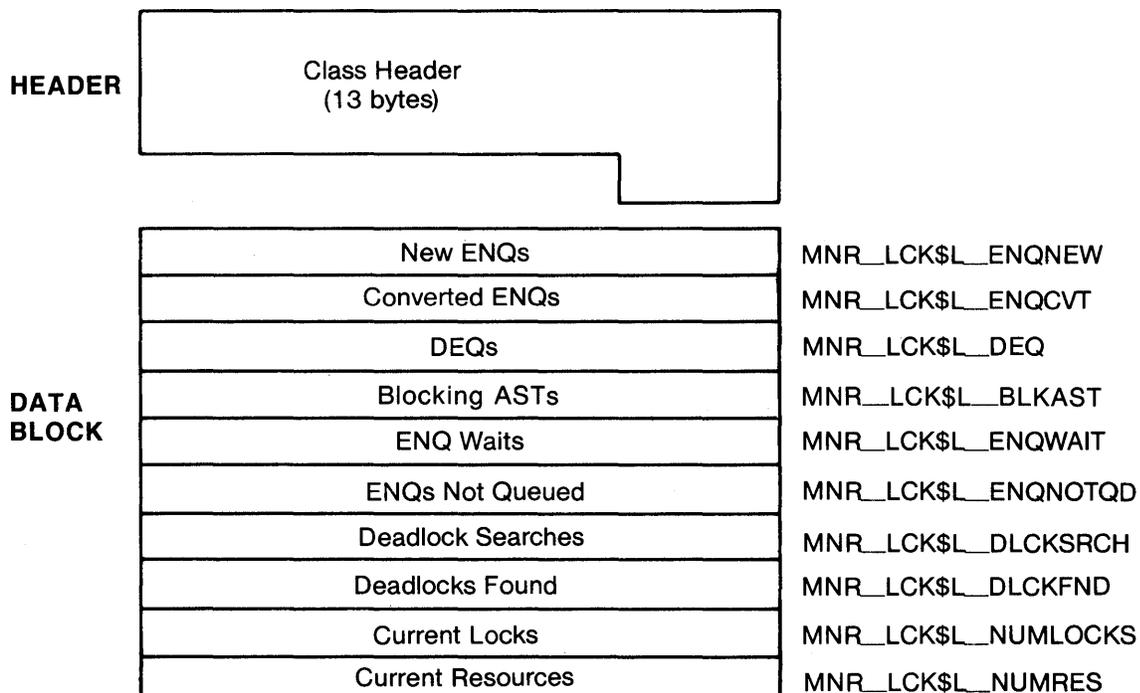
HEADER	Class Header (13 bytes)

DATA BLOCK		
	New ENQs	MNR_LCK$L_ENQNEW
	Converted ENQs	MNR_LCK$L_ENQCVT
	DEQs	MNR_LCK$L_DEQ
	Blocking ASTs	MNR_LCK$L_BLKAST
	ENQ Waits	MNR_LCK$L_ENQWAIT
	ENQs Not Queued	MNR_LCK$L_ENQNOTQD
	Deadlock Searches	MNR_LCK$L_DLCKSRCH
	Deadlocks Found	MNR_LCK$L_DLCKFND
	Current Locks	MNR_LCK$L_NUMLOCKS
	Current Resources	MNR_LCK$L_NUMRES

ZK-978-82

Table A–15 Descriptions of LOCK Record Fields

Field	Symbolic Offset	Contents
New ENQs	MNR_LCK$L_ENQNEW	Count of new ENQ (lock) requests (longword,C)
Converted ENQs	MNR_LCK$L_ENQCVT	Count of converted ENQ (lock) requests (longword,C)
DEQs	MNR_LCK$L_DEQ	Count of DEQ (unlock) requests (longword,C)
Blocking ASTs	MNR_LCK$L_BLKAST	Count of blocking ASTs queued (longword,C)
ENQ Waits	MNR_LCK$L_ENQWAIT	Count of times a lock could not be granted immediately and waited (longword,C)
ENQs Not Queued	MNR_LCK$L_ENQNOTQD	Count of times a lock could not be granted immediately and got an error status instead of waiting (longword,C)
Deadlock Searches	MNR_LCK$L_DLCKSRCH	Count of times that a deadlock search was performed (longword,C)
Deadlocks Found	MNR_LCK$L_DLCKFND	Count of times that a deadlock was found (longword,C)
Current Locks	MNR_LCK$L_NUMLOCKS	Number of locks currently in the system (longword,L)
Current Resources	MNR_LCK$L_NUMRES	Number of resources currently in the system (longword,L)

A.4.2.9   MODES Class Record

The MODES class record contains data describing time spent in each of the processor modes. The MODES class record has a record type of 2; its size depends on the number of active CPUs on the system being monitored. The size, in bytes, is calculated by adding the size of the class header, the class prefix, and the data blocks contained in the record. This is shown in the following formula (The formula assumes that all CPUs are active.):

```
13 + 8 (33 * MNR_SYI$B_MPCPUS)
```

Figure A–16 illustrates the format of the MODES class record. Table A–16 describes the fields in the data block for the MODES class record.

Figure A–16   MODES Class Record Format
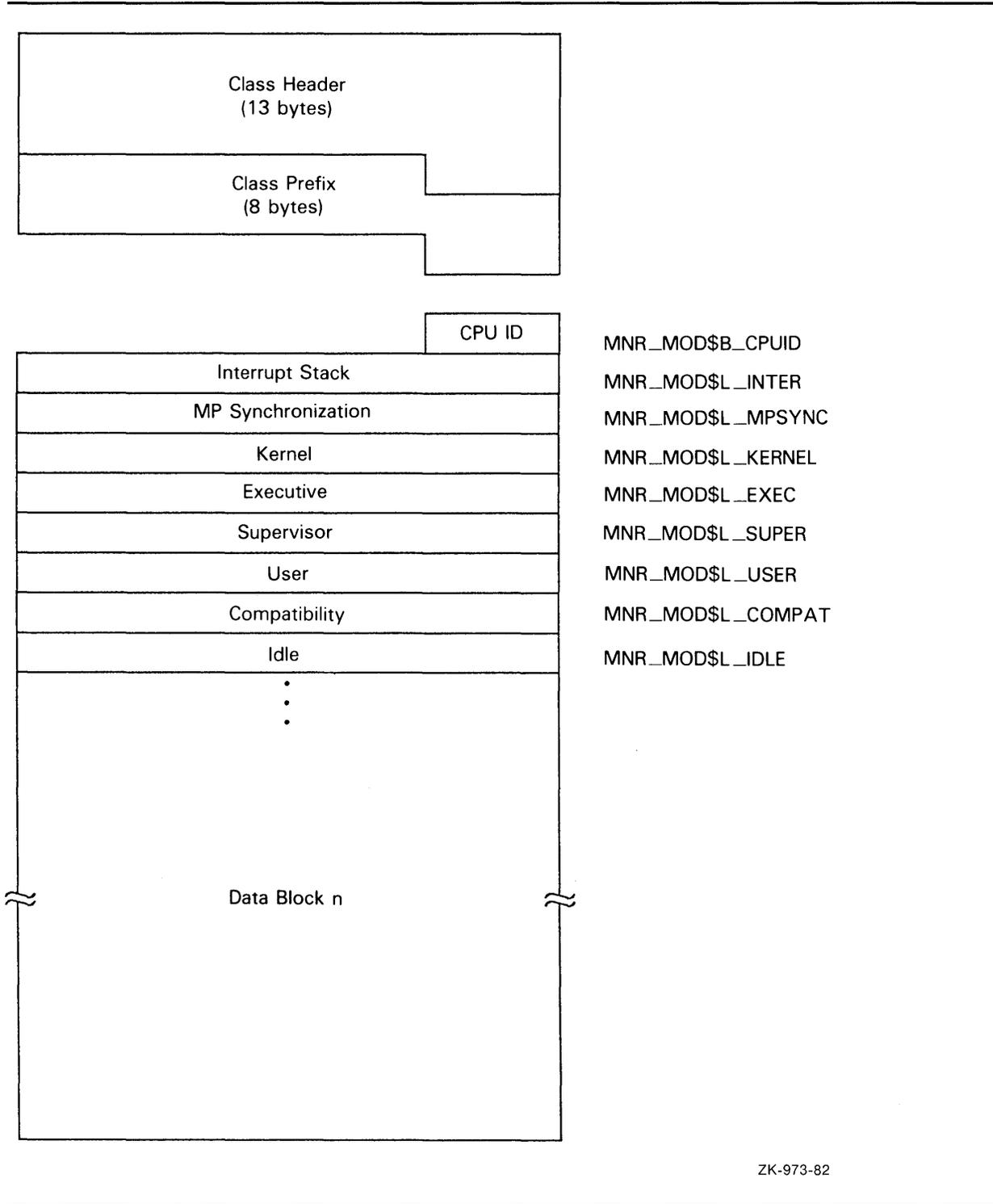
ZK-973-82

Table A-16   Descriptions of MODES Class Record Fields

Field	Symbolic Offset	Contents
CPU ID	MNR_MOD$B_CPUID	CPU identification (byte,I)
Interrupt Stack	MNR_MOD$L_INTER	Count of clock ticks (10-millisecond units) spent on interrupt stack since system was booted (longword,C)
MP Sync	MNR_MOD$L_MPSYNC	Count of clock ticks since system boot spent synchronizing multiple CPUs.
Kernel	MNR_MOD$L_KERNEL	Count of clock ticks since system boot spent in kernel mode, excluding interrupt stack time (longword,C)
Executive	MNR_MOD$L_EXEC	Count of clock ticks since system boot spent in executive mode (longword,C)
Supervisor	MNR_MOD$L_SUPER	Count of clock ticks since system boot spent in supervisor mode (longword,C)
User	MNR_MOD$L_USER	Count of clock ticks since system boot spent in user mode, excluding compatibility-mode time (longword,C)
Compatibility	MNR_MOD$L_COMPAT	Count of clock ticks since system boot spent in compatibility mode (longword,C)
Idle	MNR_MOD$L_IDLE	Count of clock ticks since system boot spent executing the NULL process (longword,C)

A.4.2.10   MSCP_SERVER Class Record

The MSCP_SERVER class record contains data describing activities of the MSCP server. The MSCP_SERVER class record has a record type of 21 and a size of 65 bytes.

Figure A-17 illustrates the format of the MSCP_SERVER class record. Table A-17 describes the fields in the data block for the MSCP class record.

Supplemental MONITOR Information—Record Formats
A.4 Class Records

Figure A-17   MSCP Class Record Format
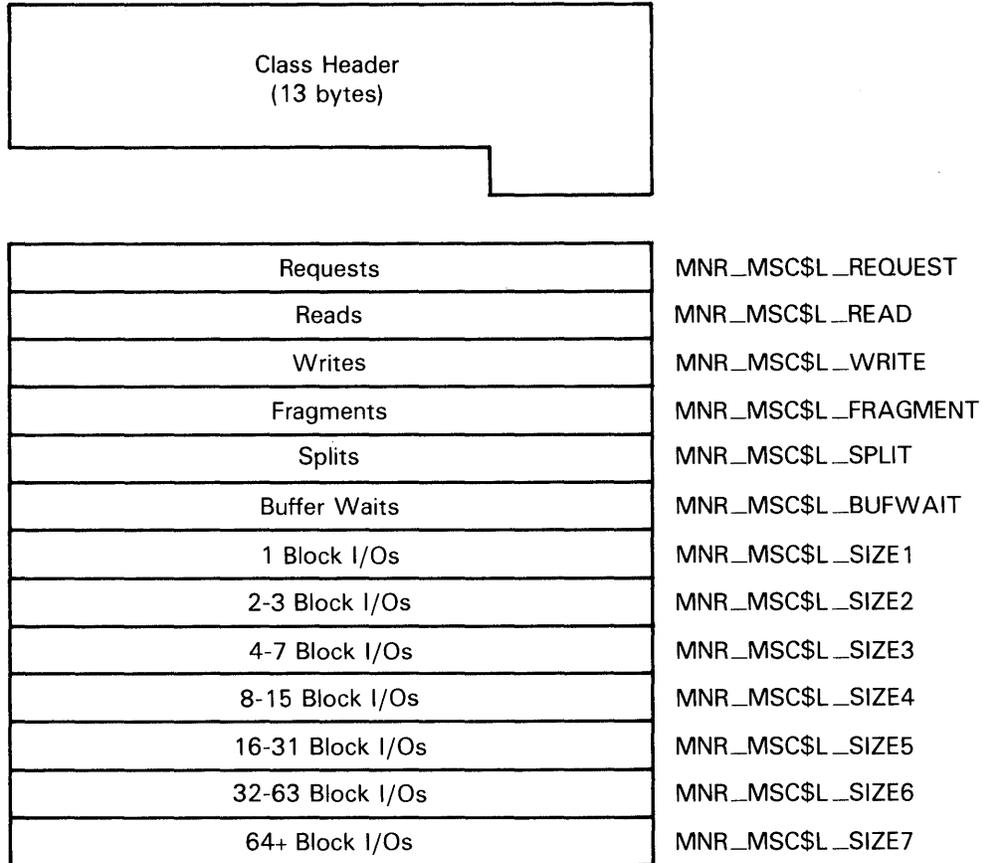
Class Header (13 bytes)	

Requests	MNR_MSC$L_REQUEST
Reads	MNR_MSC$L_READ
Writes	MNR_MSC$L_WRITE
Fragments	MNR_MSC$L_FRAGMENT
Splits	MNR_MSC$L_SPLIT
Buffer Waits	MNR_MSC$L_BUFWAIT
1 Block I/Os	MNR_MSC$L_SIZE1
2-3 Block I/Os	MNR_MSC$L_SIZE2
4-7 Block I/Os	MNR_MSC$L_SIZE3
8-15 Block I/Os	MNR_MSC$L_SIZE4
16-31 Block I/Os	MNR_MSC$L_SIZE5
32-63 Block I/Os	MNR_MSC$L_SIZE6
64+ Block I/Os	MNR_MSC$L_SIZE7

ZK-6374-HC

Table A–17 Descriptions of MSCP_SERVER Class Record Fields

Field	Symbolic Offset	Contents
Requests	MNR_MSC$L_REQUEST	Count of requests for I/O transfers by remote processors (longword,C)
Reads	MNR_MSC$L_READ	Count of requests for Read I/O transfers by remote processors (longword,C)
Writes	MNR_MSC$L_WRITE	Count of requests for Write I/O transfers by remote processors (longword,C)
Fragments	MNR_MSC$L_FRAGMENT	Count of extra fragments issued by the server (longword,C)
Splits	MNR_MSC$L_SPLIT	Count of fragmented requests issued by the server (longword,C)
Buffer Waits	MNR_MSC$L_BUFWAIT	Count of requests that had to wait for MSCP buffer memory (longword,C)
1 Block I/Os	MNR_MSC$L_SIZE1	Count of I/O requests with a length of one block (longword,C)
2—3 Block I/Os	MNR_MSC$L_SIZE2	Count of I/O requests with a length of 2 to 3 blocks (longword,C)
4—7 Block I/Os	MNR_MSC$L_SIZE3	Count of I/O requests with a length of 4 to 7 blocks (longword,C)
8—15 Block I/Os	MNR_MSC$L_SIZE4	Count of I/O requests with a length of 8 to 15 blocks (longword,C)
16—31 Block I/Os	MNR_MSC$L_SIZE5	Count of I/O requests with a length of 16 to 31 blocks (longword,C)
32—63 Block I/Os	MNR_MSC$L_SIZE6	Count of I/O requests with a length of 32 to 63 blocks (longword,C)
64+ Block I/Os	MNR_MSC$L_SIZE7	Count of I/O requests with a length equal to or greater than 64 (longword,C)

A.4.2.11   PAGE Class Record

The PAGE class record contains data describing the operation of the page management subsystem. The PAGE class record has a record type of 3 and a size of 65 bytes.

Figure A–18 illustrates the format of the PAGE class record. Table A–18 describes the fields in the data block for the PAGE class record.

Figure A–18 PAGE Class Record Format
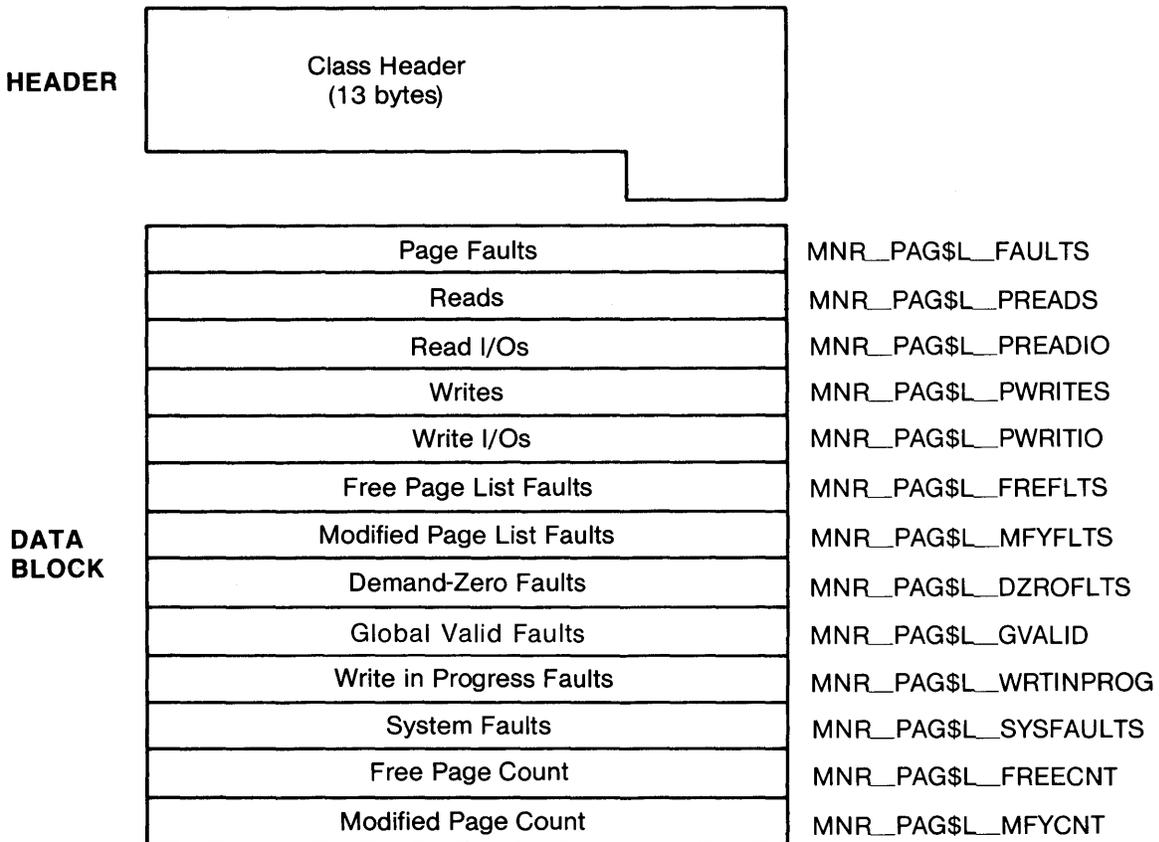
HEADER	Class Header (13 bytes)	

DATA BLOCK	Page Faults	MNR_PAG$L_FAULTS
	Reads	MNR_PAG$L_PREADS
	Read I/Os	MNR_PAG$L_PREADIO
	Writes	MNR_PAG$L_PWRITES
	Write I/Os	MNR_PAG$L_PWRITIO
	Free Page List Faults	MNR_PAG$L_FREFLTS
	Modified Page List Faults	MNR_PAG$L_MFYFLTS
	Demand-Zero Faults	MNR_PAG$L_DZROFLTS
	Global Valid Faults	MNR_PAG$L_GVALID
	Write in Progress Faults	MNR_PAG$L_WRTINPROG
	System Faults	MNR_PAG$L_SYSFAULTS
	Free Page Count	MNR_PAG$L_FREECNT
	Modified Page Count	MNR_PAG$L_MFYCNT

ZK-974-82

Table A–18 Descriptions of PAGE Class Record Fields

Field	Symbolic Offset	Contents
Page Faults	MNR_PAG$L_FAULTS	Count of page faults for all working set (longword,C)
Reads	MNR_PAG$L_PREADS	Count of pages read from disk as a result of page faults (longword,C)
Read I/Os	MNR_PAG$L_PREADIO	Count of read I/Os as a result of operations from disk page faults (longword,C)
Writes	MNR_PAG$L_PWRITES	Count of pages written to the page file (longword,C)
Write I/Os	MNR_PAG$L_PWRITIO	Count of write I/O operations to the page file (longword,C)
Free Page List Faults	MNR_PAG$L_FREFLTS	Count of pages read from the free list as a result of page faults (longword,C)
Modified Page List Faults	MNR_PAG$L_MFYFLTS	Count of pages read from the modified list as a result of page faults (longword,C)
Demand-zero Faults	MNR_PAG$L_DZROFLTS	Count of zero-filled pages allocated as a result of faults (longword,C)
Global Valid Faults	MNR_PAG$L_GVALID	Count of page faults for which the reference page was found to be valid in the system global page tables (longword,C)
Write in Progress Faults	MNR_PAG$L_WRTINPROG	Count of pages read that were in the process of being written back to disk when faulted (longword,C)
System Faults	MNR_PAG$L_SYSFAULTS	Count of page faults for which the referenced page is in system space (longword,C)
Free Page Count	MNR_PAG$L_FREECNT	Number of pages currently on free page list (longword,L)
Modified Page Count	MNR_PAG$L_MFYCNT	Number of pages currently on modified page list (longword,L)

A.4.2.12 POOL Class Record

The POOL class record contains data describing space allocation in the nonpaged dynamic pool. The POOL class record has a record type of 6 and a size of 61 bytes.

Figure A–19 illustrates the format of the POOL class record. Table A–19 describes the fields in the data block for the POOL class record.

Supplemental MONITOR Information—Record Formats
A.4 Class Records

Figure A–19   POOL Class Record Format

Class Header (13 bytes)	

Small Request Packets Free	MNR__POO$L__SRPCNT
Small Request Packets In Use	MNR__POO$L__SRPINUSE
Intermediate Request Packets Free	MNR__POO$L__IRPCNT
Intermediate Request Packets In Use	MNR__POO$L__IRPINUSE
Large Request Packets Free	MNR__POO$L__LRPCNT
Large Request Packets In Use	MNR__POO$L__LRPINUSE
Unused Kbytes	MNR__POO$L__HOLESUM
Kbytes In Use	MNR__POO$L__DYNINUSE
Unused Contiguous Space	MNR__POO$L__HOLECNT
Largest Block	MNR__POO$L__BIGHOLE
Smallest Block	MNR__POO$L__SMALLHOLE
Blocks Less Than Or Equal to 32	MNR__POO$L__SMALLCNT

ZK-1988-84

Table A–19 Descriptions of POOL Class Record Fields

Field	Symbolic Offset	Contents
Small Request Packets Available	MNR_POO$L_SRPCNT	Number of preallocated small request packets left in lookaside list (longword,L)
Small Request Packets In Use	MNR_POO$L_SRPINUSE	Number of preallocated small request packets in use (longword,L)
Intermediate Request Packets Available	MNR_POO$L_IRPCNT	Number of preallocated intermediate request packets left in lookaside list (longword,L)
Intermediate Request Packets In Use	MNR_POO$L_IRPINUSE	Number of preallocated intermediate request packets in use (longword,L)
Large Request Packets Available	MNR_POO$L_LRPCNT	Number of preallocated large request packets left in lookaside list (longword,L)
Large Request Packets In Use	MNR_POO$L_LRPINUSE	Number of preallocated large request packets in use (longword,L)
Unused Kbytes In Dynamic Nonpaged Pool	MNR_POO$L_HOLESUM	Total unused kilobytes in the dynamically allocated portion of nonpaged pool (longword,L)
Kilobytes Of Dynamic Nonpaged Pool In Use	MNR_POO$L_DYNINUSE	Number of kilobytes in the dynamically allocated portion of nonpaged pool (longword,L)
Unused Contiguous Space	MNR_POO$L_HOLECNT	Number of unused blocks of contiguous space in the dynamically allocated portion of nonpaged pool (longword,L)
Largest Block	MNR_POO$L_BIGHOLE	Size in kilobytes of the largest block of unused space in the dynamically allocated portion of nonpaged pool (longword,L)
Smallest Block	MNR_POO$L_SMALLHOLE	Size in bytes of the smallest block of unused space in the dynamically allocated portion of nonpaged pool (longword,L)
Blocks Less Than Or Equal to 32	MNR_POO$L_SMALLCNT	Number of blocks less than or equal to 32 bytes in size in the dynamically allocated portion of nonpaged pool (longword,L)

A.4.2.13 PROCESSES Class Record

The PROCESSES class record contains data describing all processes in the system. The PROCESSES class record has a record type of 0; its size depends on the number of processes being monitored. The size, in bytes, is calculated by adding the size of the class header, the class prefix, and the data blocks contained in the record. This is shown in the following formula:
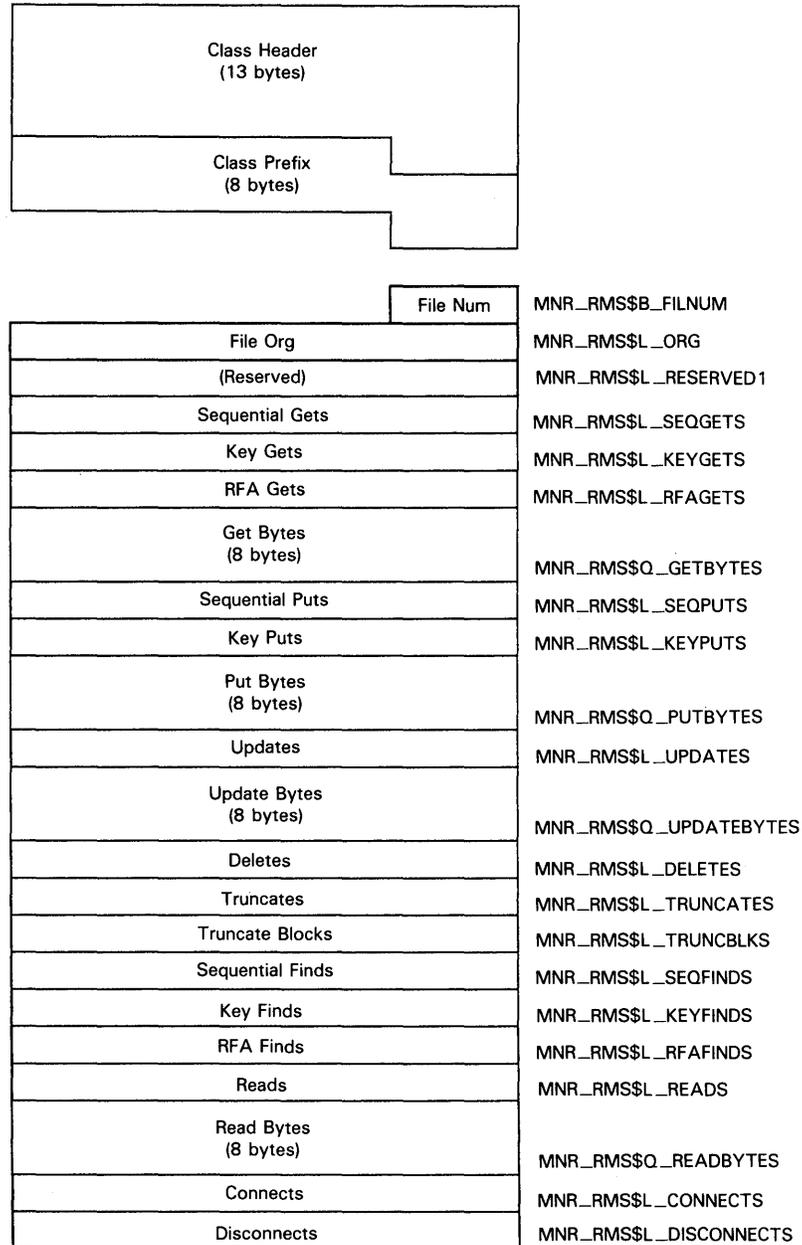
```
13 + 8 + (63 * the value of MNR_CMP$L_ELTCT)
```

Figure A–20 illustrates the format of the PROCESSES class record.
Table A–20 describes the fields in the data block for the PROCESSES class record.

Figure A–20 PROCESSES Class Record Format
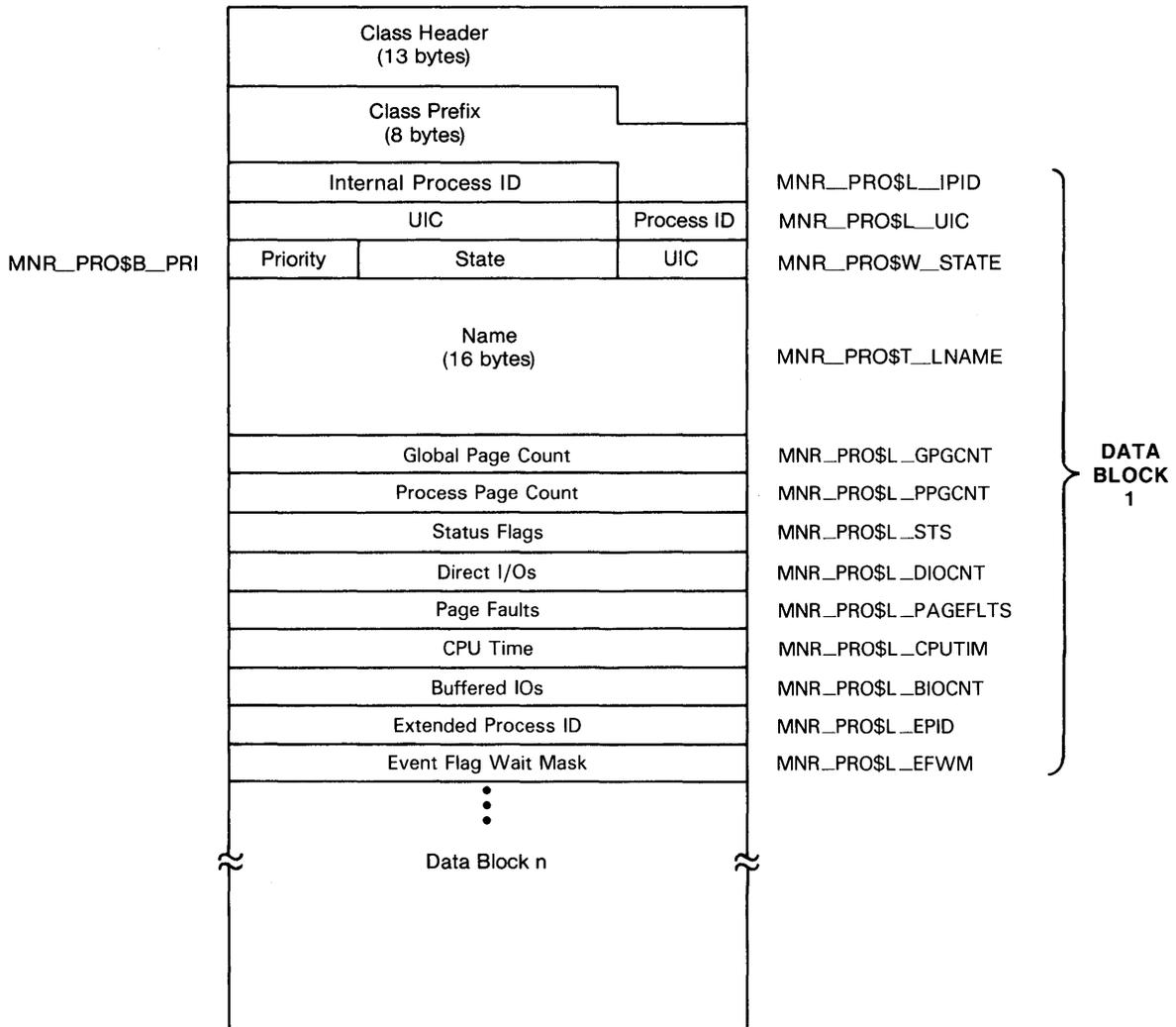
Class Header (13 bytes)	
Class Prefix (8 bytes)	
Internal Process ID	MNR_PRO$L_IPID
UIC / Process ID	MNR_PRO$L_UIC
Priority / State / UIC	MNR_PRO$W_STATE
Name (16 bytes)	MNR_PRO$T_LNAME
Global Page Count	MNR_PRO$L_GPGCNT
Process Page Count	MNR_PRO$L_PPGCNT
Status Flags	MNR_PRO$L_STS
Direct I/Os	MNR_PRO$L_DIOCNT
Page Faults	MNR_PRO$L_PAGEFLTS
CPU Time	MNR_PRO$L_CPUTIM
Buffered IOs	MNR_PRO$L_BIOCNT
Extended Process ID	MNR_PRO$L_EPID
Event Flag Wait Mask	MNR_PRO$L_EFWM

MNR_PRO$B_PRI

DATA BLOCK 1

⋮

Data Block n

ZK-971-82

Table A-20 Descriptions of PROCESSES Class Record Fields

Field	Symbolic Offset	Contents
Internal Process ID	MNR_PRO$L_IPID	Internal process identification (longword,I)
UIC	MNR_PRO$L_UIC	User identification code (Group is high-order word; Member is low-order word) (longword,I)
State	MNR_PRO$W_STATE	Current scheduling state code (word,I)
Priority	MNR_PRO$B_PRI	Current software priority (complement of 31) (byte,I)
Name	MNR_PRO$T_LNAME	Process name (counted ASCII string) (16 bytes,I)
Global Page Cnt	MNR_PRO$L_GPGCNT	Current global page count (longword,L)
Process Page Cnt	MNR_PRO$L_PPGCNT	Current process page count (longword,L)
Status Flags	MNR_PRO$L_STS	Software process status flags (PCB$V_RES bit clear implies swapped out) (longword,I)
Direct I/Os	MNR_PRO$L_DIOCNT	Direct I/O count (0 if swapped out) (longword,C)
Page Faults	MNR_PRO$L_PAGEFLTS	Page fault count (0 if swapped out) (longword,C)
CPU Time	MNR_PRO$L_CPUTIM	Accumulated CPU time, in 10ms ticks (0 if swapped out) (longword,C)
Buffered I/Os	MNR_PRO$L_BIOCNT	Buffered I/O count (0 if swapped out) (longword,C)
Extended Proc ID	MNR_PRO$L_EPID	Extended process identification (longword,I)
Event Flg Wt Mask	MNR_PRO$L_EFWM	Event flag wait mask (used for MWAITs) (longword, I)

A.4.2.14 RMS Class Record

The RMS class record contains data describing Record Management Services for specified files. The RMS class record has a record type of 20. Use the following formula to calculate the record size (The formula calculates the size by adding the size of the class header, the class prefix, and the data blocks contained in the record):

13 + 8 + (273 * MNR_CMP$L_ELTCT)

Figure A-21 illustrates the format of the RMS class record. Table A-21 describes the fields in the data block for the RMS class record.

Supplemental MONITOR Information—Record Formats
A.4 Class Records

Figure A–21 RMS Class Record Format

File Num	MNR_RMS$B_FILNUM
File Org	MNR_RMS$L_ORG
(Reserved)	MNR_RMS$L_RESERVED1
Sequential Gets	MNR_RMS$L_SEQGETS
Key Gets	MNR_RMS$L_KEYGETS
RFA Gets	MNR_RMS$L_RFAGETS
Get Bytes (8 bytes)	MNR_RMS$Q_GETBYTES
Sequential Puts	MNR_RMS$L_SEQPUTS
Key Puts	MNR_RMS$L_KEYPUTS
Put Bytes (8 bytes)	MNR_RMS$Q_PUTBYTES
Updates	MNR_RMS$L_UPDATES
Update Bytes (8 bytes)	MNR_RMS$Q_UPDATEBYTES
Deletes	MNR_RMS$L_DELETES
Truncates	MNR_RMS$L_TRUNCATES
Truncate Blocks	MNR_RMS$L_TRUNCBLKS
Sequential Finds	MNR_RMS$L_SEQFINDS
Key Finds	MNR_RMS$L_KEYFINDS
RFA Finds	MNR_RMS$L_RFAFINDS
Reads	MNR_RMS$L_READS
Read Bytes (8 bytes)	MNR_RMS$Q_READBYTES
Connects	MNR_RMS$L_CONNECTS
Disconnects	MNR_RMS$L_DISCONNECTS

ZK-6108/1-HC

Figure A–21 Cont'd. on next page

Figure A–21 (Cont.) RMS Class Record Format

Extends	MNR_RMS$L_EXTENDS
Extend Blocks	MNR_RMS$L_EXTBLOCKS
Flushes	MNR_RMS$L_FLUSHES
Rewinds	MNR_RMS$L_REWINDS
Writes	MNR_RMS$L_WRITES
Write Bytes (8 bytes)	MNR_RMS$Q_WRITEBYTES
File Lock ENQs	MNR_RMS$L_FLCKENQS
File Lock DEQs	MNR_RMS$L_FLCKDEQS
File Lock Conversions	MNR_RMS$L_FLCKCNVS
Local Buffer ENQs	MNR_RMS$L_LBLCKENQS
Local Buffer DEQs	MNR_RMS$L_LBLCKDEQS
Local Buffer Conversions	MNR_RMS$L_LBLCKCNVS
Global Buffer ENQs	MNR_RMS$L_GBLCKENQS
Global Buffer DEQs	MNR_RMS$L_GBLCKDEQS
Global Buffer Conversions	MNR_RMS$L_GBLCKCNVS
Global Section ENQs	MNR_RMS$L_GSLCKENQS
Global Section DEQs	MNR_RMS$L_GSLCKDEQS
Global Section Conversions	MNR_RMS$L_GSLCKCNVS
Record Lock ENQs	MNR_RMS$L_RLCKENQS
Record Lock DEQs	MNR_RMS$L_RLCKDEQS
Record Lock Conversions	MNR_RMS$L_RLCKCNVS
Append Lock ENQs	MNR_RMS$L_APPLCKENQS
Append Lock DEQs	MNR_RMS$L_APPLCKDEQS
Append Lock Conversions	MNR_RMS$L_APPLCKCNVS
File Lock Blocking ASTs	MNR_RMS$L_FLBLKASTS
Local Buffer Blocking ASTs	MNR_RMS$L_LBLBLKASTS
Global Buffer Blocking ASTs	MNR_RMS$L_GBLBLKASTS
Append Lock Blocking ASTs	MNR_RMS$L_APPBLKASTS
Local Cache Hits	MNR_RMS$L_LCACHEHITS
Local Cache Attempts	MNR_RMS$L_LCACHEATT
Global Cache Hits	MNR_RMS$L_GCACHEHITS
Global Cache Attempts	MNR_RMS$L_GCACHEATT
Global Buffer Read I/Os	MNR_RMS$L_GBRDIRIOS
Global Buffer Write I/Os	MNR_RMS$L_GBWDIRIOS
Local Buffer Read I/Os	MNR_RMS$L_LBRDIRIOS
Local Buffer Write I/Os	MNR_RMS$L_LBWDIRIOS

ZK-6108/2-HC

Figure A–21 (Cont.) RMS Class Record Format

Bucket Splits	MNR_RMS$L_BKTSPLT
Multi Bucket Splits	MNR_RMS$L_MBKTSPLT
Opens	MNR_RMS$L_OPENS
Closes	MNR_RMS$L_CLOSES
(Reserved)	MNR_RMS$L_RESERVED2
(Reserved)	MNR_RMS$L_RESERVED3

. . .

Data Block n

ZK-6108/3-HC

Table A–21 Descriptions of RMS Record Fields

Field	Symbolic Offset	Contents
File Number	MNR_RMS$B_FILNUM	Sequential number of the file (byte,I)
Organization	MNR_RMS$L_ORG	Organization of the file (longword,I)
Reserved	MNR_RMS$L_RESERVED1	Reserved (longword)
Sequential GETs	MNR_RMS$L_SEQGETS	Count of sequential $GETs to the file (longword,C)
Key GETs	MNR_RMS$L_KEYGETS	Count of keyed $GETs to the file (longword,C)
RFA GETs	MNR_RMS$L_RFAGETS	Count of $GETs by record-file-address to the file (longword,C)
GET Bytes	MNR_RMS$Q_GETBYTES	Total number of bytes required for all $GETs issued (quadword,C)
Sequential PUTs	MNR_RMS$L_SEQPUTS	Count of sequential $PUTs to the file (longword,C)
Key PUTs	MNR_RMS$L_KEYPUTS	Count of keyed $PUTs to the file (longword,C)

Table A–21 (Cont.)   Descriptions of RMS Record Fields

Field	Symbolic Offset	Contents
PUT Bytes	MNR_RMS$Q_PUTBYTES	Total number of bytes required for all $PUTs issued (quadword,C)
UPDATEs	MNR_RMS$L_UPDATES	Count of $UPDATES to the file (longword,C)
UPDATE Bytes	MNR_RMS$Q_UPDATEBYTES	Total number of bytes required for all $UPDATEs issued (quadword,C)
DELETEs	MNR_RMS$L_DELETES	Count of $DELETEs to the file (longword,C)
TRUNCATEs	MNR_RMS$L_TRUNCATES	Count of $TRUNCATEs to the file (longword,C)
TRUNCATE Blocks	MNR_RMS$L_TRUNCBLKS	Total blocks required for all $TRUNCATEs issued (longword,C)
Sequential FINDs	MNR_RMS$L_SEQFINDS	Count of sequential $FINDs to the file (longword,C)
Key FINDs	MNR_RMS$L_KEYFINDS	Count of keyed $FINDs to the file (longword,C)
RFA FINDs	MNR_RMS$L_RFAFINDS	Count of $FINDs by record-file-address to the file (longword,C)
READs	MNR_RMS$L_READS	Count of $READs to the file (longword,C)
READ Bytes	MNR_RMS$Q_READBYTES	Total bytes required for all $READs to the file (quadword,C)
CONNECTs	MNR_RMS$L_CONNECTS	Count of $CONNECTs to the file (longword,C)
DISCONNECTs	MNR_RMS$L_DISCONNECTS	Count of $DISCONNECTs to the file (longword,C)
EXTENDs	MNR_RMS$L_EXTENDS	Count of $EXTENDS to the file (longword,C)
EXTEND Blocks	MNR_RMS$L_EXTBLOCKS	Total blocks required for all EXTENDs to the file (longword,C)
FLUSHes	MNR_RMS$L_FLUSHES	Count of $FLUSHes to the file (longword,C)
REWINDs	MNR_RMS$L_REWINDS	Count of $REWINDs to the file (longword,C)
WRITES	MNR_RMS$L_WRITES	Count of $WRITESs to the file (longword,C)
WRITE Bytes	MNR_RMS$Q_WRITEBYTES	Total bytes required for all $WRITEs issued (quadword,C)
File Lock ENQs	MNR_RMS$L_FLCKENQ	Count of file lock ENQs to the file (longword,C)
File Lock DEQs	MNR_RMS$L_FLCKDEQ	Count of file lock DEQs to the file (longword,C)
File Lock Conversions	MNR_RMS$L_FLCKCONVS	Count of file lock conversions for the file (longword,C)

Supplemental MONITOR Information—Record Formats

A.4 Class Records

Table A–21 (Cont.)   Descriptions of RMS Record Fields

Field	Symbolic Offset	Contents
Local Buffer ENQs	MNR_RMS$L_LBLCKENQS	Count of local buffer ENQs to the file (longword,C)
Local Buffer DEQs	MNR_RMS$L_LBLCKDEQS	Count of local buffer DEQs to the file (longword,C)
Local Buffer Conversions	MNR_RMS$L_LBLCKCNVS	Count of local buffer conversions for the file (longword,C)
Global Buffer ENQs	MNR_RMS$L_GBLCKENQS	Count of global buffer ENQs to the file (longword,C)
Global Buffer DEQs	MNR_RMS$L_GBLCKDEQS	Count of global buffer DEQs to the file (longword,C)
Global Buffer Conversions	MNR_RMS$L_GBLCKCNVS	Count of global buffer conversions for the file (longword,C)
Global Section ENQs	MNR_RMS$L_GSLCKENQS	Count of global section ENQs to the file (longword,C)
Global Section DEQs	MNR_RMS$L_GSLCKDEQS	Count of global section DEQs to the file (longword,C)
Global Section Conversions	MNR_RMS$L_GSLCKCNVS	Count of global section conversions for the file (longword,C)
Record Lock ENQs	MNR_RMS$L_RLCKENQS	Count of record lock ENQs to the file (longword,C)
Record Lock DEQs	MNR_RMS$L_RLCKDEQS	Count of record lock DEQs to the file (longword,C)
Record Lock Conversions	MNR_RMS$L_RLCKCNVS	Count of record lock conversions for the file (longword,C)
Append Lock ENQs	MNR_RMS$L_APPLCKENQS	Count of append lock ENQs to the file (longword,C)
Append Lock DEQs	MNR_RMS$L_APPLCKDEQS	Count of append lock DEQS to the file (longword,C)
Append Lock Conversions	MNR_RMS$L_APPLCKCNVS	Count of append lock conversions for the file (longword,C)
File Lock Blocking ASTs	MNR_RMS$L_FLBLKASTS	Count of file lock blocking ASTs for the file (longword,C)
Local Buffer Blocking ASTs	MNR_RMS$L_LBLBLKASTS	Count of local buffer blocking ASTs for the file (longword,C)
Global Buffer Blocking ASTs	MNR_RMS$L_GBLBLKASTS	Count of global buffer blocking ASTs for the file (longword,C)
Append Lock Blocking ASTs	MNR_RMS$L_APPBLKASTS	Count of apppend lock blocking ASTs for the file (longword,C)
Local Cache Hits	MNR_RMS$L_LCACHEHITS	Count of local cache hits for the file (longword,C)
Local Cache Attempts	MNR_RMS$L_LCACHEATT	Count of local cache attempts for the file (longword,C)
Global Cache Hits	MNR_RMS$L_GCACHEHITS	Count of global cache hits for the file (longword,C)

Table A–21 (Cont.) **Descriptions of RMS Record Fields**

Field	Symbolic Offset	Contents
Global Cache Attempts	MNR_RMS$L_GCACHEATT	Count of global cache attempts (longword,C)
Global Buffer Read I/Os	MNR_RMS$L_GBRDIRIOS	Count of global buffer read I/Os for the file (longword,C)
Global Buffer Write I/Os	MNR_RMS$L_GBWDIRIOS	Count of global buffer write I/Os for the file (longword,C)
Local Buffer Read I/Os	MNR_RMS$L_LBRDIRIOS	Count of local buffer read I/Os for the file (longword,C)
Local Buffer Write I/Os	MNR_RMS$L_LBWDIRIOS	Count of local buffer write I/Os for the file (longword,C)
Bucket Splits	MNR_RMS$L_BKTSPLT	Count of bucket splits for the file (longword,C)
Multibucket Splits	MNR_RMS$L_MBKTSPLT	Count of multibucket splits for the file (longword,C)
Opens	MNR_RMS$L_OPENS	Count of the times the file was opened (longword,C)
Closes	MNR_RMS$L_CLOSES	Count of the times the file was closed (longword,C)
Reserved	MNR_RMS$L_RESERVED2	Reserved (longword)
Reserved	MNR_RMS$L_RESERVED3	Reserved (longword)

A.4.2.15 SCS Class Record

The SCS class record contains data describing SCS (System Communication Services) activity for all SCS connections in the system, on a per-node basis. The SCS class record has a record type of 15; its size depends on the number of nodes being monitored. The size, in bytes, is calculated by adding the size of the class header, the class prefix, and the data blocks contained in the record. This is shown in the following formula:

```
13 + 8 + (56 * the value of MNR_CMP$L_ELTCT)
```

Figure A–22 illustrates the format of the SCS class record. Table A–22 describes the fields in the data block for the SCS class record.

Supplemental MONITOR Information—Record Formats
A.4 Class Records

Figure A–22 SCS Class Record Format

HEADER	Class Header (13 bytes)
PREFIX	Class Prefix (8 bytes)

NODE NAME	MNR__SCS$T__NODENAME

Datagrams Sent	MNR__SCS$L__DGSENT	
Datagrams Received	MNR__SCS$L__DGRCVD	
Datagrams Discarded	MNR__SCS$L__DGDISCARD	
Sequenced Messages Sent	MNR__SCS$L__MSGSENT	
Sequenced Messages Received	MNR__SCS$L__MSGRCVD	**DATA BLOCK 1**
Block Transfer Send-datas	MNR__SCS$L__SNDATS	
Kilobytes Sent by Send-datas	MNR__SCS$L__KBYTSENT	
Block Transfer Request-datas	MNR__SCS$L__REQDATS	
Kilobytes Received by Request-datas	MNR__SCS$L__KBYTREQD	
Block Transfer Kilobytes Mapped	MNR__SCS$L__KBYTMAPD	
Connections Queued for Send Credit	MNR__SCS$L__QCRCNT	
Connections Queued for Buffer Descriptor	MNR__SCS$L__QBDTCNT	

DATA BLOCKS

DATA BLOCK n

ZK-1670-84

A–42

Table A–22 Descriptions of SCS Class Record Fields

Field	Symbolic Offset	Contents
Node Name	MNR_SCS$T_NODENAME	Name of remote cluster node (counted ASCII string) (8 bytes,I)
Datagrams Sent	MNR_SCS$L_DGSENT	Count of datagrams sent to the remote node (longword,C)
Datagrams Received	MNR_SCS$L_DGRCVD	Count of datagrams received from the remote node (longword,C)
Datagrams Discarded	MNR_SCS$L_DGDISCARD	Count of datagrams discarded by the CI port driver (longword,C)
Sequenced Msgs Sent	MNR_SCS$L_MSGSENT	Count of sequenced messages sent to the remode node (longword,C)
Seqenced Msgs Received	MNR_SCS$L_MSGRCVD	Count of sequenced messages received from the remote node (longword,C)
Block Transfer Send-datas	MNR_SCS$L_SNDATS	Count of block transfer send-datas initiated on the local node, targeted for the remote node (longword,C)
Kilobytes Sent by Send-datas	MNR_SCS$L_KBYTSENT	Count of kilobytes sent as a result of send-datas (longword,C)
Block Transfer Request-datas	MNR_SCS$L_REQDATS	Count of block transfer request-datas initiated on the local node, targeted for the remote node (longword,C)
Kilobytes Received by Request-datas	MNR_SCS$L_KBYTREQD	Count of kilobytes received as a result of request-datas (longword,C)
Block Transfer Kilobytes Mapped	MNR_SCS$L_KBYTMAPD	Count of kilobytes mapped for block transfers (longword,C)
Connections Queued For Send Credit	MNR_SCS$L_QCRCNT	Count of times connections are queued for send credits (longword,C)
Connections Queued For Buffer Descriptor	MNR_SCS$L_QBDTCNT	Count of times connections are queued for buffer descriptors (longword,C)

A.4.2.16 STATES Class Record

The STATES class record contains data describing the number of processes in each of the scheduler states. The STATES class record has a record type of 1 and a size of 69 bytes.

Figure A–23 illustrates the format of the STATES class record. Table A–23 describes the fields in the data block for the STATES class record.

Figure A–23 STATES Class Record Format

HEADER	Class Header (13 bytes)	

DATA BLOCK		
	Collided Page Wait	MNR_STA$L_COLPG
	Misc. Resource Wait	MNR_STA$L_MWAIT
	Common Event Flag	MNR_STA$L_CEF
	Page Fault Wait	MNR_STA$L_PFW
	Local Event Flag, Inswapped	MNR_STA$L_LEF
	Local Event Flag, Outswapped	MNR_STA$L_LEFO
	Hibernate, Inswapped	MNR_STA$L_HIB
	Hibernate, Outswapped	MNR_STA$L_HIBO
	Suspended, Inswapped	MNR_STA$L_SUSP
	Suspended, Outswapped	MNR_STA$L_SUSPO
	Free Page Wait	MNR_STA$L_FPG
	Compute State, Inswapped	MNR_STA$L_COM
	Compute State, Outswapped	MNR_STA$L_COMO
	Current	MNR_STA$L_CUR

ZK-972-82

Table A–23 Descriptions of STATES Class Record Fields

Field	Symbolic Offset	Contents
Collided Page Wait	MNR_STA$L_COLPG	Number of processes in collided page wait (longword,L)
Misc Resource Wait	MNR_STA$L_MWAIT	Number of processes in miscellaneous resource wait (longword,L)
Common Event Flag Wait	MNR_STA$L_CEF	Number of processes in common event flag wait (longword,L)
Page Fault Wait	MNR_STA$L_PFW	Number of processes in page fault wait (longword,L)
Loc Event Flag, Inswapped	MNR_STA$L_LEF	Number of processes in local event flag wait, inswapped (longword,L)
Loc Event Flag, Outswapped	MNR_STA$L_LEFO	Number of processes in local event flag wait, outswapped (longword,L)
Hibernate Inswapped	MNR_STA$L_HIB	Number of processes in hibernate wait, inswapped (longword,L)
Hibernate Outswapped	MNR_STA$L_HIBO	Number of processes in hibernate wait, outswapped (longword,L)
Suspended Inswapped	MNR_STA$L_SUSP	Number of processes in suspended wait, inswapped (longword,L)
Suspended Outswapped	MNR_STA$L_SUSPO	Number of processes in suspended wait, outswapped (longword,L)
Free Page Wait	MNR_STA$L_FPG	Number of processes in free wait (longword,L)
Compute State Inswapped	MNR_STA$L_COM	Number of processes in compute state, inswapped (longword,L)
Compute State Outswapped	MNR_STA$L_COMO	Number of processes in compute state, outswapped (longword,L)
Current	MNR_STA$L_CUR	Number of current processes (longword,L)

A.4.2.17 SYSTEM Class Record

The SYSTEM class record contains data describing the overall operation of the three major system components (CPU, memory, I/O). The SYSTEM class record has a record type of 17 and a size of 49 bytes. Note that when the SYSTEM class is recorded, the PROCESSES, STATES, and MODES classes are also recorded, even if not explicitly requested.

Figure A–24 illustrates the format of the SYSTEM class record. Table A–24 describes the fields in the data block for the SYSTEM class record.

Supplemental MONITOR Information—Record Formats

A.4 Class Records

Figure A—24 SYSTEM Class Record Format

Class Header (13 bytes)	

CPU Busy	MNR_SYS$L_BUSY
Other States	MNR_SYS$L_OTHSTAT
Process Count	MNR_SYS$L_PROCS
Page Faults	MNR_SYS$L_FAULTS
Read I/Os	MNR_SYS$L_PREADIO
Free Page Count	MNR_SYS$L_FREECNT
Modified Page Count	MNR_SYS$L_MFYCNT
Direct I/Os	MNR_SYS$L_DIRIO
Buffered I/Os	MNR_SYS$L_BUFIO
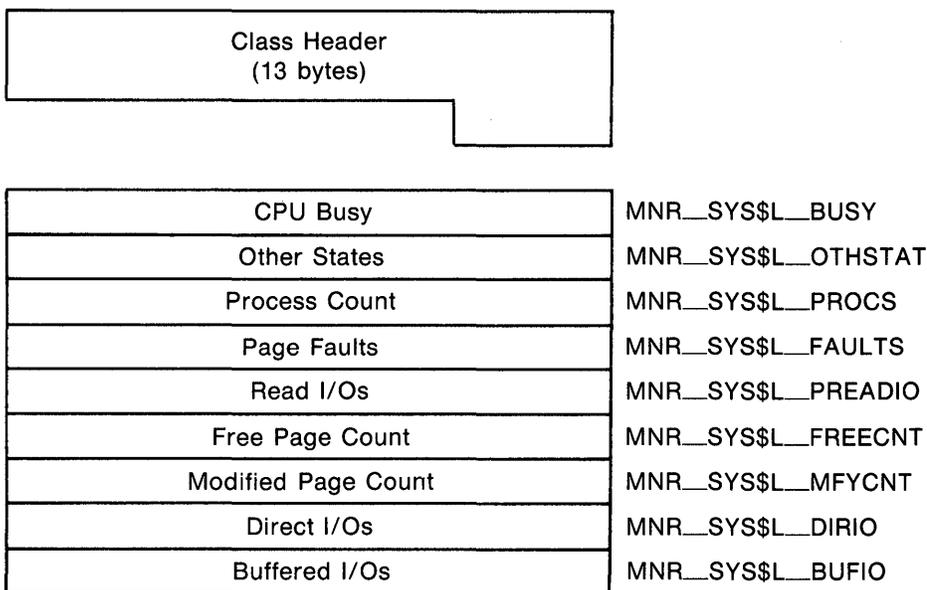
ZK-1986-84

Table A—24 Descriptions of SYSTEM Class Record Fields

Field	Symbolic Offset	Contents
CPU Busy	MNR_SYS$L_BUSY	Count of clock ticks (10-millisecond units) spent in all CPU modes since system was booted (longword,C)
Other States	MNR_SYS$L_OTHSTAT	Number of processes in states other than LEF, LEFO, HIB, HIBO, COM, COMO, PFW, and MWAIT (longword,L)
Process Count	MNR_SYS$L_PROCS	Number of processes in system (longword,L)
Page Faults	MNR_SYS$L_FAULTS	Count of page faults for all working sets (longword,C)
Read I/Os	MNR_SYS$L_PREADIO	Count of read I/Os resulting from disk page faults (longword,C)
Free Page Count	MNR_SYS$L_FREECNT	Number of pages currently on free page list (longword,L)
Modified Page Count	MNR_SYS$L_MFYCNT	Number of pages currently on modified page list (longword,L)
Direct I/Os	MNR_SYS$L_DIRIO	Count of direct I/O operations (longword,C)
Buffered I/Os	MNR_SYS$L_BUFIO	Count of buffered I/O operations (longword,C)

Index

B

C

D

E

F

H

I

Index

Reader's Comments

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

I would like to see more/less _____

_____

What I like best about this manual is _____

_____

What I like least about this manual is _____

_____

I found the following errors in this manual:

Page   Description

_____   _____

_____   _____

_____   _____

_____   _____

_____   _____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

I am using **Version** _____ of the software this manual describes.

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Phone _____

digital™

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01–3/J35 110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

Reader's Comments

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

I would like to see more/less _____

_____

_____

What I like best about this manual is _____

_____

_____

What I like least about this manual is _____

_____

_____

I found the following errors in this manual:

Page   Description

_____   _____

_____   _____

_____   _____

_____   _____

_____   _____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

I am using **Version** _____ of the software this manual describes.

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

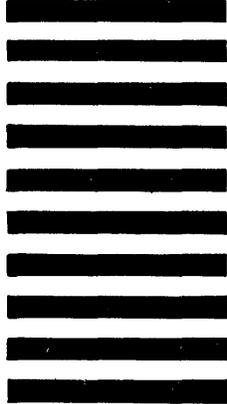_____ Phone _____

d i g i t a l™

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01–3/J35 110 SPIT BROOK ROAD
NASHUA, NH 03062-9987