SPM-11M REFERENCE MANUAL

V1

OPERATING SYSTEM AND VERSION: RSX-11M V3.2

SOFTWARE VERSION: SPM-11M V1.0

Software Services,
digital equipment corporation
maynard, massachusetts

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

CONTENTS

PREFACE

MANUAL OBJECTIVES

The SPM-11M Reference Manual is a manual describing the structure and use of the SPM-11M software supported by DIGITAL on RSX-11M operating systems.

INTENDED AUDIENCE

This manual is for all users of SPM-11M.

STRUCTURE OF THIS DOCUMENT

Chapter 1 is an overview of the architecture of SPM-11M.

Chapter 2 describes the components of SPM-11M.

Chapter 3 describes how to use SPM-11M.

Chapter 4 describes data collection and reduction procedures.

Chapter 5 describes how to read the performance report.

Chapter 6 delineates the resources used by SPM-11M.

Chapter 7 is the usage notes for SPM-11M.

Appendix A is a specification of the Hook Records.

Appendix B is a specification of the Monitor Parameter Block (MPB).

Appendix C is a specification of the Information Records.

Appendix D is a specification of the System Metrics Record.

Appendix E is a specification of the logfile format.

Appendix F is a sample performance report.

Appendix G is a sample SMPGEN.CMD procedure.

Chapter 1
OVERVIEW OF SPM-11M

The Software Performance Monitor (SPM-11M) is a software instrument used to measure the performance of a computer system running the RSX-11M operating system.

SPM-11M is a natural counterpart to RSX-11M: while RSX-11M manages the resources of the computer system, SPM-11M remembers how the resources were used over a period of time, similar to the way a tape recorder remembers a set of sound patterns. Moreover, SPM-11M has the capability of "playing back" these recordings of resource usage, so an analyst can study how and when the resources were used.

SPM-11M is an event driven performance monitor. As each event of interest occurs, the performance monitor is called.

1.1 RESOURCES

The key resources of interest are:

- CPU
- Main memory
- I/O devices
- File system
- Task loader

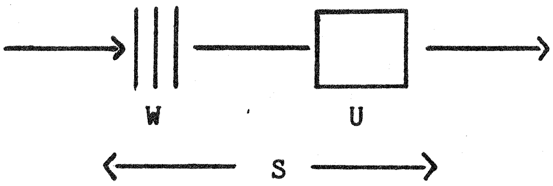
Note that a resource can be a hardware resource -- such as the CPU, main memory or an I/O device; or a software resource -- such as the file system or the task loader.

1.2 USERS

A user is the entity which uses the resources managed by the operating system. We define a user to be an RSX-11M task. Of primary concern is the workload the task places on the system, that is, how much and for how long the resources are used. To describe a task's workload we will define a set of metrics for resource usage.

1.3 RESOURCE MODEL

Because a number of tasks may concurrently request the use of a resource, and only one task may use the resource at a time, a queue will form for the resource. A simple queued resource is depicted as follows:



The vertical bars represent the queued resource requests and the square represents the resource. Three intervals are shown on the diagram. "W" denotes the interval during which a task is waiting for the resource. "U" denotes an interval when a task is using the resource. "S" is the sum of "W" and "U", called the service time.

1.4 EVENTS

The activity of the RSX-11M operating system can be traced by defining a set of key events and recording a set of raw information associated with each event. The events of interest are those which capture the metrics of resource usage.

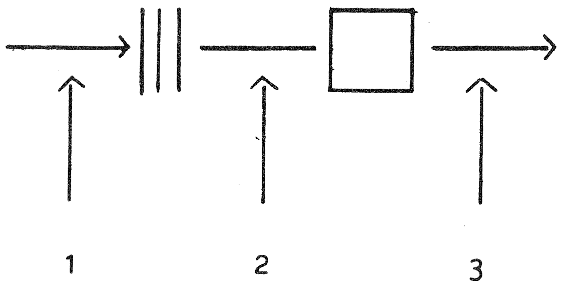
For example, we may want to know how long a task waits for the disk, how many times it uses the disk and how long it uses the disk. The following events are defined to capture the I/O activity:

1. Queue I/O packet to driver

2. Driver gets I/O packet

3. Driver posts I/O completion

Applied to the queued resource model for the disk, these I/O events appear as follows:



The complete set of events defined for RSX-11M is specified in Appendix A.

1.5 TASK METRICS

The task metrics quantify the workload that individual tasks place on the system.

The task metrics are computed for all three types of intervals: U, W and S. The performance report will contain a line of task metrics for each resource used by the task. The task metrics are:

- Count

  A simple metric for resource usage is a count of the number of times the resource was used during the life of a task. For example, if a task does four QIOs to a disk, the I/O request count is four.

- Total Time

  Another simple metric is how long the task uses a resource. For example, if a task did four QIOs and the I/O requests took 20ms, 60ms, 10ms and 50ms to complete, the total usage time would be 140ms.

- % Resident Time

  Also of importance is what percentage the total usage time is of the time the task was in memory. For example, if the previous task was in memory for one second the percentage of resident time spent using the disk was 140ms/1000ms *100% = 14%.

- Interval Statistics

  The statistical nature of the usage time is of key importance.

  The metrics defined for interval statistics are:

  - Minimum interval
  - Mean (average) interval
  - Maximum interval
  - Coefficient of Variation

  The coefficient of variation is the standard deviation of the intervals divided by the mean interval.

  The coefficient of variation is of particular importance because it tells whether the data is clustered about the mean (small c.v.) or widely scattered about the mean (large c.v.). In fact, if the c.v. is large, the mean may tell very little about the nature of the data, since very small values and very large values may combine to produce a mean which is not representative of any real value.

1.6 SYSTEM METRICS

The task workload metrics defined above are useful for determining who is using the system resources and how much each user is consuming.

Also of importance is the performance of the system as a whole. To this purpose we define a set of system metrics.

The system metrics vary as a function of time. To see how a given system metric varies with time, histograms are computed for the metric. Each histogram describes the per cent usage of a resource for one sampling interval. The sampling interval is a settable parameter described in Chapter 3, section 3.2.1.

CPU

The CPU can be in one of three general states:

- User
- Kernel (system)
- Idle

A useful metric is the per cent of time spent in each state, that is:

- % time in user state
- % time in kernel state
- % time in idle state

This gives us a general picture of how effectively the CPU is being utilized (% time in user state + % time in kernel state), as well as the overhead imposed by the operating system (% time in kernel state).

Memory

On RSX-11M the memory space is divided into partitions. The user can select one system-controlled partition for monitoring by SPM-11M. The performance monitor will compute the per cent of the partition that was in use at the end of each sampling interval and print a histogram.

The user will most commonly select the partition "GEN" for monitoring.

I/O Devices

The user can select one I/O device to be monitored by SPM-11M for system-wide usage. SPM-11M will compute the percent of time the device was utilized for each sampling interval and print a histogram. If the device is mounted Files-11, SPM-11M will compute the percent disk space used at the end of each sampling interval and print a histogram.

Pool

The RSX-11M system pool contains the dynamically allocated data structures used by the executive. SPM-11M will compute the per cent of pool that was in use at the end of each sampling interval and print a histogram.

Checkpoint file(s)

SPM-11M will monitor checkpoint files allocated by the ACS command. The performance monitor will compute the per cent of all active checkpoint file space in use at the end of each sampling interval and print a histogram.

Chapter 2
THE COMPONENTS OF SPM-11M

The SPM-11M performance monitor consists of two main components: the Data Collection Component and the Data Reduction Component. The Data Collection Component records the resource usage events of the operating system to a logfile; the Data Reduction Component reads the logfile and produces a performance report suitable for analysis.

This general architecture is depicted in Figure 1.

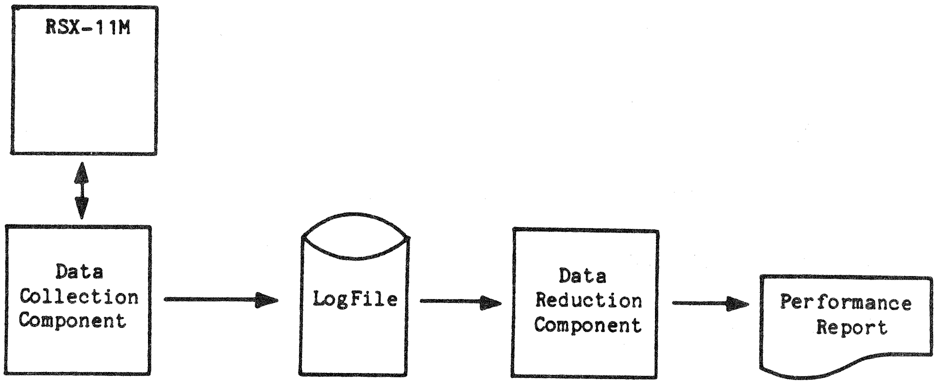


Figure 1

By decoupling the Data Collection Component from the Data Reduction Component in time, the performance monitor can collect detailed information about a huge number of events, albeit in a raw format. The Data Reduction Component can then be used to selectively analyze and filter the information in the logfile to produce a performance report. This architecture shifts the computational aspects of the performance monitor from the Data Collection Component to the Data Reduction Component. The benefit derived is that the performance monitor uses less CPU time, thus minimizing its interference with the CPU. The cost is more I/O activity, which in most systems can be directed to a secondary mass storage device not in the path of the major stream of I/O activity.

2.1 DATA COLLECTION COMPONENT

The Data Collection Component consists of both software and hardware distributed in various places within an RSX-11M system.

The basic pieces of the Data Collection Component are:

- Hooks
- Hook Dispatcher
- Hook Routines
- Hook Records
- KW11-P Clock and Handler
- Hook Buffers
- Logfile
- Data Collection Task (...COL)
- Termination Task (...TRM)
- COL Message Task (COLMSG)

The relationship between these pieces is shown in Figure 2.

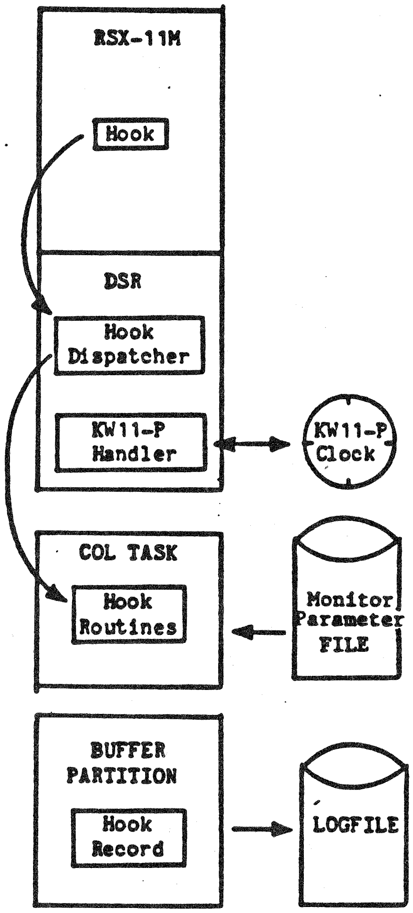


Figure 2

Heuristic Procedure for Data Collection

At this point a heuristic procedure for data collection will be given, although the terms used in the procedure have not been precisely defined.

When a resource usage event occurs in RSX-11M, the corresponding hook calls the hook dispatcher, which uses the hook ID to call the appropriate hook routine. The hook routine collects information relevant to the hook and assembles it into a hook record, which resides in a hook buffer. When a hook buffer is filled with hook records, it is written to the logfile. The logfile is closed at the end of the measurement interval, and subsequently reopened by the Data Reduction Component. It reads the hook records and uses them to derive the resource usage metrics and print a performance report.

The details of these pieces will now be developed more fully.

2.1.1 Hooks

A hook is simply an implementation technique for capturing a resource usage event. For SPM-11M, a hook is implemented as a subroutine call inserted at the point in the operating system where the event occurs. The subroutine call passes control and one argument, the hook ID, to the hook dispatcher. The form of an SPM-11M hook is as follows:

```
CALL @$HKDSP
.WORD H$XXXX
```

where XXXX is mnemonic identifying the particular hook. A complete specification of all hooks is given in Appendix A.

2.1.2 Hook Dispatcher

Hooks are resident within the operating system; their method of insertion is addressed in Chapter 3, section 3.1.3. However, most of the code and database for the Data Collection Component is not resident within the RSX-11M executive.

The bulk of the data collection code is resident in the data collection task called COL (taskname is ...COL). The chief benefit derived from this design is that only a small amount of executive space (which translates directly to the critical resource: pool) is used by the performance monitor.

The purpose of the hook dispatcher is to map and call the hook routines which are resident in the COL task. Note that, though the hook routines are resident in the task partition for COL, they are executed in the context of a directive or interrupt service routine. The hook dispatcher is resident in pool, is allocated when COL is activated, and is deallocated when COL exits. The hook dispatcher is analagous to an ICB for a loadable driver.

A one word dispatch vector labelled $HKDSP is added to system common (SYSCM). This word points to the entry point in the hook dispatcher resident in pool. It is the port through which all hooks pass control from,the RSX-11M executive to the Data Collection Component of the performance monitor.

2.1.3 Hook Routine

A hook routine copies information which is generally scattered about the executive (typically in SYSCM or SYSTB) into a hook record which is resident in a static common or dynamic region. A timestamp is also recorded in the hook record.

There are two general types of hooks: task and system. The task hooks collect data relevant to resources used by tasks; the system hooks collect data which is system-wide (e.g., % CPU time in user, kernel and idle).

2.1.4 Hook Record

The information collected at each hook is formatted into a hook record. The hook record provides an identical representation of a hook's information content both in memory and in the logfile.

Each hook record contains both generic and hook-specific data. Generic data applies to all hooks. Generic data consists of:

- Hook ID
- Taskname
- Timestamp
- Task's TI (UCB address)

Hook specific data applies to only one or a small subset of the hooks. For example, the H$CDRP (call directive processor) hook records the directive ID code as hook-specific data.

Appendix A specifies the information content of the hook records.

2.1.5 KW11-P Clock and Handler

For precise performance measurements, SPM-11M needs a high speed clock running independent of the RSX-11M system clock. It is therefore a requirement that a KW11-P clock be installed on a machine running SPM-11M. This KW11-P clock must be in addition to the RSX-11M system clock, be it a KW11-L or KW11-P.

The KW11-P is initialized by the COL task to run at 100 kHz (10 microsecond ticks) and to interrupt every 65536 ticks (approximately every 0.65 second).

The KW11-P clock time is maintained as as 2 word (32 bit) value which, at a rate of 100 kHz, will overflow in a little over 8 hours.

To preclude the possibility of overflow of the KW11-P clock time, a maximum measurement interval of 8 hours has been established. The task COL enforces this by setting an 8 hour mark-time when activated. If the timer expires, COL will stop data collection and exit.

The COL task allocates a piece of system pool and links it to the KW11-P clock vector. This pool node serves as the clock interrupt handler, which updates the high order clock time located in SYSCM at $PCLKH. The low order clock time is not stored in memory; rather, each hook routine copies the current contents of the KW11-P clock's count register to the hook record directly to get the low order time word.

2.1.6 Hook Buffer

Hook records are stored in buffers which are resident in either a static common or a dynamic region. When a hook buffer is full, the hook routine sets an event flag which signals the COL task to write the buffer to the logfile.

The hook buffers are linked together into a circular list. There may be from 2 to 8 buffers in a list, and all the buffers must be of the same length. A buffer may be from 256 words to 4096 words long.

If the buffers are resident in a static common, the maximum size common which can be used is 4096 words. This limitation is imposed because the COL task uses one APR to map the static common.

If the buffers are resident in a dynamic region, the region can be from 256 words to 32K words long.

2.1.7 Logfile

The logfile contains four types of records:

- Monitor Parameter Block (MPB)
- Information Records
- Hook Records
- System Metric Records

See Appendix E for a specification of the logfile.

Monitor Parameter Block

The Monitor Parameter Block (MPB) is used by the COL task to initialize its measurement parameters. The MPB is formatted as an Information Record with record type I$MPB. A detailed description of the MPB is given in Appendix B.

The MPB is generated by a command file called SMPGEN.CMD. See Chapter 3, section 3.2.1 for details.

Information Records

The Information Records are used to supplement the Hook Records in the logfile with system context useful to the Data Reduction Component.

When COL starts up it reads the MPB from the MPB file and writes it to the logfile. Following the MPB, COL writes a stream of Information Records which contain a number of RSX-11M data structures defining the initial system state. These data structures are:

- Task Control Block (TCB)
- Partition Control Block (PCB)
- Device Control Block (DCB)
- Unit Control Block (UCB)

Each data structure is copied to a separate Information Record.

When COL begins a measurement interval (starts data collection), it records the RSX-11M system time as well as the SPM-11M time (KW11-P time) to an Information Record (I$TMR).

When COL ends a measurement interval (stops data collection) it copies the RSX-11M system time, SPM-11M time, CPU time accrued in kernel state and CPU time accrued in idle state to an Information Record (I$STM).

A complete specification of all Information Records is given in Appendix C.

Hook Records

As each hook point is encountered in the executive, a Hook Record is written to the logfile. See section 2.1.4 for details.

System Metrics Record

At the end of each sample interval, a System Metrics Record is written to the logfile. This record contains information which describes the system-wide usage of the selected resources.

The System Metrics Record is specified in Appendix D.

Logfile Media

The logfile can be resident on disk as a Files-11 file or on magtape as a sequence of tape records.

If the logfile is resident on disk, it consists of a sequence of virtual blocks (virtual block=256 words). The records within these virtual blocks are defined according to the record definitions for the MPB, Information Records, Hook Records and System Metric Records. Note that the logfile does not have any other record structure (such as FCS or RMS type records).

If the logfile is on a magtape, the MPB, Information Records and Hook Records are imbedded in a stream of tape records. Each tape record is equal in length to a hook buffer.

2.1.8 Data Collection Task (...COL)

This section gives a functional description of the COL task. A guide to using COL is provided in Chapter 3, section 3.2.2.

As noted before, this task contains both code which executes in executive context (as a directive or interrupt process) and code which executes in the context of a privileged task. The code which executes at executive level is the hook routines.

2.1.8.1 COL Executive Level Code -

The hook routines were made a part of the COL task for a number of reasons:

- They must be resident in some partition outside of executive space.
- They can easily be loaded as part of the COL task.
- They implement routines which are common to hooks and to the COL task.

2.1.8.2 COL Task Level Code -

Most of the COL task consists of code which is executed at task level. The major functions of the COL task are:

- Initialization
- Starting data collection
- Writing to logfile
- Stopping data collection
- Cleanup

2.1.8.2.1 Initialization -

When COL begins execution, there is very little SPM-11M code elsewhere in the system. All that exists are the hooks in the executive which all call a dummy return subroutine in SYSXT. It is the job of COL to initialize all code and databases required for the SPM-11M performance monitor. The specific sequence of initialization functions performed by COL are:

- Read MPB file and store MPB in COL buffer.
- Create dynamic region (if necessary) and set up ring of hook buffers.
- Write MPB and Information Records (RSX-11M data structures) to logfile.
- Mark COL task as fixed in memory and nonshufflable.
- Initialize hooks specified in MPB.

- Allocate pool node for hook dispatcher and copy code to pool node.

- Allocate pool node for KW11-P clock handler and copy code to pool node. Link pool node to KW11-P clock vector.

- Initialize KW11-P clock and turn it on.

- Initialize measurement parameters as specified in MPB.

2.1.8.2.2 Starting Data Collection -

When data collection is disabled, the hook dispatch vector $HKDSP points to a dummy routine in SYSXT which simply steps over the hook ID and returns to the instruction in the executive following the hook.

Just prior to starting data collection, COL writes a "start data collection" Information Record (I$TMR) to the hook buffer.

To enable data collection the COL task simply points the hook dispatch vector $HKDSP to the hook dispatcher code resident in pool.

2.1.8.2.3 Writing Hook Records to Logfile -

The steady state function of the COL task is to write Hook Records to the logfile. When a hook routine fills a hook buffer, it sets an event flag for COL which wakes COL from its wait state. COL then writes the hook buffer to disk or magtape (waiting for completion of the write), and loops to see if another hook buffer is ready to write. When no more hook buffers are outstanding, COL returns to the wait state.

If the hook buffers fill up too rapidly for COL to write them to the logfile, COL will set the overflow status bit. If the overflow status is set when a measurement interval expires, COL will print a status message on the console device (CO:). In general, the data collected during a measurement interval should not be used for data reduction if the overflow condition occurred.

2.1.8.2.4 Stopping Data Collection -

To stop data collection, COL merely points the hook dispatch vector $HKDSP to the dummy routine at $HKEXT in SYSXT. It then writes a stop data collection Information Record (I$STM) to the hook buffer. Any outstanding records in the hook buffer will be written to the logfile.

2.1.8.2.5 Cleanup -

Before COL exits, it must release any system resources it has allocated and deactivate any hardware it has turned on. The following actions are thus performed:

- Turn off KW11-P clock.
- Deallocate pool nodes for hook dispatcher and KW11-P clock handler.
- Unfix COL task.

2.1.9 Termination Task (...TRM)

The purpose of the termination task (...TRM) is to signal the COL task to stop data collection and exit. TRM provides a way to manually terminate the COL task, as opposed to automatic shutoff by timeout or buffers (see Chapter 3, section 3.2.1).

2.1.10 COL Message Task (COLMSG)

The COL message task contains the ASCII text for the error messages issued by the COL task. If the COLMSG task is installed and a COL error occurs, the full error message text will be printed. If a COL error occurs and the COLMSG task is not installed, the COL task will print an error number which can be correlated with the correct error message in Chapter 3, section 3.2.2.

2.2 DATA REDUCTION COMPONENT

The main purpose of the Data Reduction Component is to read the raw data from the logfile (hook records) and produce a performance report detailing the resources used by the various tasks in the system.

The Data Reduction Component accomplishes this in a multi-step process involving two tasks:

- Monitor Data Reduction (MDR) Task
- DRS Task

There are three major steps in the data reduction process:

1. Map Hook Records to Task Workload Records and generate system-wide metrics section of performance report (MDR Pass 1).

2. Sort Task Workload Records (DRS).

3. Summarize Task Workload Records and generate task metrics section of performance report (MDR Pass 2).

The Task Workload Records are intermediate data structures which contain the data collected for each task and each resource used by the task.

This general procedure is depicted in Figure 3.

DATA REDUCTION COMPONENT

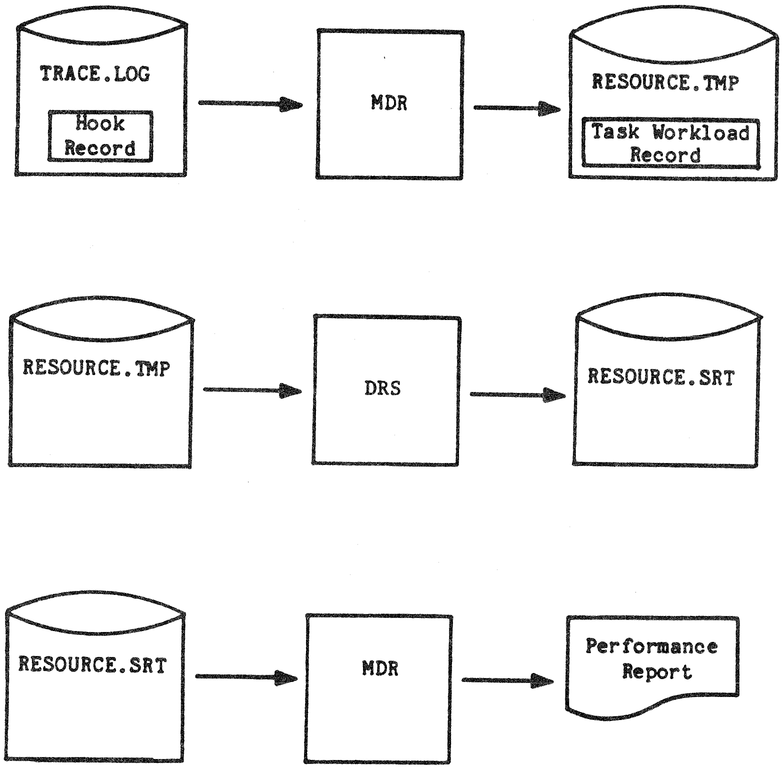


Figure 3

From the figure we see there are four databases which are processed by the two tasks:

- TRACE.LOG (logfile containing Hook Records).
- RESOURCE.TMP (intermediate file containing unsorted Task Workload Records).
- RESOURCE.SRT (intermediate file containing sorted Task Workload Records.
- REDUCE.RPT (final performance report).

2.2.1 MDR Pass 1

MDR reads hook records from the logfile (default name is TRACE.LOG) and maps the information to the Task Workload Records in the intermediate file RESOURCE.TMP.

2.2.2 DRS Pass

DRS sorts the Task Workload Records using the following three keys:

- Generic Taskname
- Task's TI UCB address
- Task start time.

NOTE: By "generic taskname" we mean the installed taskname of a multiuser task. For example, the generic taskname for the task PIPT5 is ...PIP. If the task is not multiuser, generic taskname is simply the run time taskname.

The three keys listed above are concatenated into one "big" key, with the generic taskname forming the highest order part of the key, the Task's TI UCB address forming the middle order part of the key, and the Task start time forming the low order part of the key. The DRS task sorts all the Task Workload Records using this one big key. With this technique the DRS task only needs to do one sort operation on the file RESOURCE.TMP.

When DRS is done sorting the Task Workload Records, the records will be arranged as follows:

- All records with the same generic taskname will be clustered together.
- Within a generic taskname cluster, all records with the same TI will be clustered together.
- All records within a taskname-TI cluster will be sequenced by ascending task start time.

DRS writes the sorted Task Workload Records to the file RESOURCE.SRT.

2.2.3 MDR Pass 2

MDR reads the Task Workload Records from RESOURCE.SRT and generates a performance report. A description of the performance report is given in Chapter 5.

Chapter 3
GUIDE TO USING SPM-11M

3.1 INSTALLATION

3.1.1 SPM-11M Kit

The SPM-11M kit is distributed on 800 BPI magtape. It contains all the files necessary to install and use SPM-11M. The files are in DOS tape format. When copied to disk, the files will occupy approximately 800 disk blocks.

The command file PMONCPY.CMD can be used to copy the kit files from tape to tape, tape to disk, disk to tape or disk to disk. All files are contained in the directory [11,2]. The contents of the kit are:

Component	File
Kit copy command file	PMONCPY.CMD
SPM-11M mini-reference guide	SPM.DOC
MPB GEN command file	SMPGEN.CMD
Macro definition prefix file	HKMAC.MAC
Executive source correction command file	PMONSLP.CMD
Executive source correction files	*.COR
F11ACP patch command file	PMONPAT.CMD
F11ACP patch source	DISPAT.PAT
COL/TRM build command file	PMONBLD.CMD
COL overlay descriptor file	COLBLD.ODL
COL/TRM object library	COL.OLB
MDR object library (EIS version)	MDREIS.OLB
MDR object library (FPP version)	MDRFPP.OLB
MDR/CLF build command file	MDRTKB.CMD
Data reduction sort task	DRS.TSK
DRS map file	DRS.MAP
DRS sort command file	DRS.CMD
CLF task	CLF.TSK
CLF map file	CLF.MAP

3.1.2 Prerequisites to Installing SPM-11M

COL Prerequisites

The following are the prerequisites for running the COL task:

- Mapped System

  SPM-11M can only be installed on a mapped RSX-11M system.

- Get Partition Parameters Directive

  The COL task will only work if support for the Get Partition Parameters (GPRT$) directive is generated into the executive.

- AST Support

  The COL task will collect System Metrics only if AST support is generated into the executive.

- Send and Receive

  The COL task will issue full ASCII error message text via the error message task COLMSG if support for Send and Receive is generated into the executive.

- PLAS Support

  If PLAS support is not generated into the executive, the COL task buffering is limited to a maximum of a 4K word static common.

  If PLAS support is generated into the executive, the COL task buffering may be in a dynamic region up to 32K words long.

MDR Prerequisites

MDR requires the following features of the system where it is run:

- EIS instructions

- support for the executive directives: ASTX$, EXTK$, GPRT$, GTSK$.

- checkpointing to a dynamically allocated system checkpoint file.

- any features needed by the SYSLIB routines CSI$, GCML$, and FCS.

3.1.3 Installing SPM-11M

To install SPM-11M, you must start with:

- An RSX-11M V3.2 distribution kit
- An RSX-11M autopatch kit at the current level
- An SPM-11M distribution kit

To install SPM-11M you must perform an RSX-11M SYSGEN. A SYSGEN is required because source code for SPM-11M is inserted into the RSX-11M sources. SPM-11M source code is added to the following RSX-11M source modules:

SPM-11M Function	RSX-11M Modules
SPM-11M Macros and Definitions	RSXMCO,RSXMC
Hooks	DRDSP,DRQIO,DREIF,DREXP, DRREG,REQSB,SYSXT,IOSUB, LOADR
Additional Stack Space	LOWCR
Common Database	SYSCM
Dummy Hook Return Routine	SYSXT

The SPM-11M installation procedure consists of the following steps:

- Copy RSX-11M V 3.2 kit to baseline disk (labeled "RSXM26").
- Perform autopatch to current level.
- Assign SY: to the "RSXM26" disk.
- Choose a directory to which you want the SPM-11M kit files copied. This will be your SPM-11M kit directory.

- Assuming the SPM-11M directory is [g,m]

```
>SET /UIC=[g,m]

>UFD SY:[g,m]

>FLX =ddn:[11,2]PMONCPY.CMD

>@PMONCPY
                    .
                    .
                    .
APPLY CORRECTIONS TO EXECUTIVE SOURCES? Y
                    .
                    .
                    .
APPLY PATCH TO F11ACP DISPATCHER MODULE (DISPAT)? Y

@<EOF>

>@[200,200]SYSGEN

>@[200,200]SYSGEN2
```

NOTE: If the patch to F11ACP fails (checksum error), SPM-11M will only calculate service times for F11ACP. If the patch succeeds, SPM-11M will generate usage, wait and service times.

If PLAS is not generated into the executive, create a static common up to 4K words long with VMR to provide space for the COL buffers.

If you <u>do not</u> elect to correct the executive sources or patch F11ACP via PMONCPY.CMD you can do so later via the command files:

1) @PMONSLP for executive source corrections

2) @PMONPAT for F11ACP patch

- When the SYSGEN completes, perform the following steps:

```
>SET /UIC=[g,m]

>@PMONBLD
```

NOTE: Be sure to taskbuild COL using the baseline library [1,1]SYSLIB.OLB (without ANSI or big buffering FCS routines). Otherwise COL will be greater than 4096 words and will not be able to map its buffers via APR6.

You can build the MDR or CLF tasks separately from PMONBLD.CMD by invoking the command file MDRTKB.CMD.

- You are now ready to run @SMPGEN to generate an MPB and run COL. This procedure is described in the next section.

3.2 OPERATION

3.2.1 Generating the Monitor Parameter Block (MPB)

The MPB is the structure which contains the measurement parameters.

To create an MPB you invoke the command file SMPGEN.CMD.

See Appendix G for an example SMPGEN.

The procedure is as follows:

```
>SET /UIC=[g,m]

>@SMPGEN
```

where [g,m] is your SPM-11M kit directory.

The MPB is created in the following way: SMPGEN creates a source file (default filespec is TRACE.PAR), assembles and builds the code in the file to produce a binary MPB file (default filespec is TRACE.MPB). It is this file which is read by the data collection task COL to initialize the measurement parameters.

In addition to the files mentioned above, SMPGEN also will output, if requested, a saved answer file (default filespec is TRACE.CMD). This file can be input to another run of SMPGEN and will initialize the symbol table to the values of the variables saved in the file.

SMPGEN is driven from a menu of parameter categories. To display the menu, type an <ESC>. The menu appears as:

```
;
; LEGAL PARAMETER CATEGORIES ARE :
;        HOOK            TASK            MODE            LABEL
;        BUFFER          DEVICE          SYSTEM          DONE
;
```

The parameter categories can be specified in any order. If a parameter category is not specified, the parameters will receive default values from either:

- The saved answer file, if one is specified.
- The table of default parameters.

Table of Default Parameters

HOOKS - all hooks enabled.

BUFFERS - DYNAMIC region named TRCPAR created in partition GEN, 2 buffers with 8 blocks per buffer.

TASKS - All tasks enabled for data collection.

DEVICES - All devices enabled for data collection.

MODE - MANUAL mode.

SYSTEM - System Metrics enabled with 60 second sampling interval.

LABEL - No label.

To select a parameter category, enter the name of the parameter category desired as shown in the following example:

```
ENTER PARAMETER CATEGORY? HOOK
```

Once a parameter category is selected, the detailed parameters are specified. Each parameter category has a menu of responses which are possible for that parameter category. To get the menu, type <ESC>. When you are finished entering parameters, type "DONE".

A detailed description of each parameter category follows.

- Label Field

  To select the label parameter category, type: LABEL

  The following query will be displayed:

  ```
  ENTER MPB LABEL?
  ```

  The response may be any string from 0 to 63 characters long. This user-definable string is stored in the MPB and will typically contain notes and comments applicable to a particular measurement run. It is printed as part of the heading on each page of the performance report.

- Hooks

  To select the hooks parameter category, type: HOOK

  The hooks determine which resource usage events are recorded to the logfile.

  You do not select each hook individually. Rather, you select one or more groups of hooks. Each group of hooks contains the subset of hooks which will generate a consistent set of data for measuring the usage of a particular resource. Hooks may be chosen from the following groups:

  - CPU USAGE HOOKS

    This hook group measures task CPU usage.

    It is required for all metrics for task CPU usage.

  - MEMORY USAGE HOOKS

    This hook group measures task memory usage.

    It is required for all metrics for task memory usage and % resident time.

  - I/O DEVICE USAGE HOOKS

    This hook group measures task usage of I/O devices and the file system. The file system is regarded as any ACP which manages a device. Examples are F11ACP and MTAACP.

This hook group is required for all metrics relative to task usage of I/O devices or the file system.

- Buffers

To select the buffers parameter category, type: BUFFER.

The hook buffers may be resident in either a static common or a dynamic region. The buffer type is specified by:

```
ENTER BUFFER TYPE?
```

The response may be either STATIC indicating a static common or DYNAMIC indicating a dynamic region.

NOTE: If "STATIC" is selected, the COL task must have been linked to a static common in the installation phase (see Chapter 3, section 3.1.3).

If "DYNAMIC" is selected, COL must not be linked to the static common.

If the buffer type is DYNAMIC, the name of the system-controlled partition in which to create the dynamic region is specified by:

```
ENTER NAME OF BUFFER MAIN (SYS) PARTITION?
```

The name of the buffer common/region is specified by:

```
ENTER NAME OF BUFFER COMMON/REGION?
```

The partition will be divided up into a number of hook buffers, determined by:

```
ENTER NUMBER OF BUFFERS [2.:8.:2.]?
```

Note that from 2 to 8 buffers may be specified.

Finally, the length of each buffer is specified by:

```
ENTER BUFFER SIZE IN 512 B BLOCKS?
```

A buffer can be from 1 to 16 blocks long; in other words, from 256 words to 4096 words long.

If a static common is used for buffering, the total space required for buffers must fit into the common. If a dynamic region is used, COL will create a region large enough to contain the buffers.

- Tasks

SPM-11M can be set up to collect data for specific tasks. For example, there may be some tasks whose performance metrics are not desired at all. Typically, system "service" tasks, such as F11ACP, ...MCR and QMG... fall into this category.

To select the <u>task</u> parameter category, type: TASK.

The following query will appear:

```
DO YOU WANT TO TRACE ALL TASKS?
```

If your answer is N, you will be given two choices for selecting the tasks for data collection:

- Specify the tasks not to trace.

- Specify the tasks to trace.

The choice is made from the query:

```
DO YOU WANT TO SPECIFY THE TASKS NOT TO TRACE?
```

If the answer is Y, you are prompted with:

```
ENTER NAME OF TASK NOT TO TRACE?
```

If the answer is N, you are prompted with:

```
ENTER NAME OF TASK TO TRACE?
```

You then type, one per line, the names of the tasks desired or not desired. Each taskname is from 1 to 6 characters long. To terminate the list of tasks, type <CR>.

Up to 64 tasknames can be specified.

- Devices

To get the devices parameter category, type: DEVICE.

SPM-11M can be set up to collect data for specific classes of devices. If you want to collect data only for selected classes of devices, answer N to the following query:

```
DO YOU WANT TO COLLECT DATA FOR ALL DEVICES?
```

A device in RSX-11M may fall into one of four classes depending on the bit setting of the "Device Characteristics Word" U.CW1 in the UCB.

- Directory Device (DV.DIR)
- Sequential Device (DV.SQD)
- Terminal Device (DV.TTY)
- All Other Devices

To enable data collection for a device class, answer Y to the query for that device class. The queries are:

```
TRACE DISK (DIRECTORY DEVICE) I/O ACTIVITY?

TRACE TAPE (SEQUENTIAL DEVICE) I/O ACTIVITY?

TRACE TERMINAL I/O ACTIVITY?

TRACE ALL OTHER DEVICES I/O ACTIVITY?
```

- Mode

  To select the mode parameter category, type: MODE.

  The performance monitor can be set up to collect data in three different modes. You select the mode by answering:

  ```
  ENTER MODE?
  ```

  The possible modes are: MANUAL, AUTOSTOP and REPEAT. Each mode determines the manner in which a measurement interval begins and ends. A description of each mode follows:

  - Manual

    In manual mode, a measurement interval begins when the COL task is invoked and ends when the TRM (termination) task is run.

- Autostop

  In autostop mode, a measurement interval begins when the COL task is invoked and ends when either of the following two events occur:

  - A mark-time expires.

  - A predetermined number of buffers have been filled.

  You determine the type of autostop by responding to:

  ```
  ENTER SAMPLING TYPE?
  ```

  with either TIME or BUFFER.

  If TIME is specified, you then have to specify the timeout interval:

  ```
  ENTER SAMPLING INTERVAL TIME UNIT [D:MIN]

  ENTER SAMPLING INTERVAL IN units ?
  ```

  where "units" is the time unit specified in the previous query. Legal units are SEC, MIN or HR.

  NOTE: Do not confuse this sampling interval with the system metrics sampling interval described in the next section.

  If BUFFER is specified as the interval type, you must specify the number of buffers per sample:

  ```
  ENTER NUMBER OF BUFFERS PER SAMPLE?
  ```

  Note that each buffer is from 1 to 16 disk blocks (256 words) long.

- Repeat

  Repeat mode is autostop mode repeated a number of times. The sampling interval is specified as for autostop mode. In addition, you must specify:

  - The number of intervals

  - The length of time <u>between</u> intervals.

You specify these parameters by answering the following queries:

```
ENTER MAX. NUMBER OF SAMPLES?

ENTER INTERVAL BETWEEN SAMPLES TIME UNIT [D:MIN]?

ENTER INTERVAL BETWEEN SAMPLES IN units ?
```

where "units" is the time unit specified in the previous query.

- System

  To select the system-metrics parameter group, type "SYSTEM".

  If you want the system metrics, answer Y the the next query:

  ```
  DO YOU WANT THE SYSTEM METRICS? [Y/N]:
  ```

  If you answered Y, you must then specify the sampling interval by:

  ```
  ENTER SYSTEM METRICS SAMPLING INTERVAL (SEC) ?
  ```

- Done

  When you are finished entering parameters, type "DONE". The SMPGEN procedure will then create:

  - A monitor parameter block (MPB) file (filename.MPB).

  - A parameter source file (filename.PAR).

  - A saved answer file (filename.CMD, if specified).

  Note: SMPGEN will prompt you for the parameter source file and saved answer file, and you must enter a valid file specification.

3.2.2 Using COL

The COL task is used to initialize the performance monitor, start data collection, collect data, stop data collection and clean up the performance monitor before exiting.

Installing COL

The COL task can be installed in any partition. If possible, create a separate 4K word partition for COL. If installed in a system-controlled partition, COL will automatically fix itself and mark itself as non-shufflable. It is therefore desirable to install and run COL when there is little or no system activity in the partition in which COL is installed so as to load COL in the lower end of the partition and reduce fragmentation of the partition. The COL task will occupy approximately 4K words of memory.

COL is taskbuilt with a default priority of 230. Note that COL has a higher priority than F11ACP and ...MCR, but a lower priority than ...LDR. It is highly recommended that the priority of COL not be altered to a value lower than the default value of 230, otherwise excessive overflows may occur which will invalidate the measurement runs.

If full ASCII message text output is desired, install the COLMSG task.

If you are using a static common for COL buffers, be sure to install it before installing the COL task.

Install COL and COLMSG by:

```
        >INS TRCPAR          ;ONLY FOR STATIC COMMON

        >INS COL

        >INS COLMSG
```

Running COL

To run COL, type a command line of the form:

```
        >COL logfile/sw=MPBfile/sw
```

The default logfile specification is SY:[cur uic]TRACE.LOG. The default MPBfile specification is SY:[cur uic]TRACE.MPB.

Stopping COL

There are two ways to terminate COL:

- RUN TRM
- Specify AUTOSTOP or REPEAT mode in SMPGEN.

RUN TRM will terminate COL regardless of the mode in which COL is running: MANUAL, AUTOSTOP or REPEAT.

COL will also terminate if any of the following abnormal conditions are detected:

- Powerfail recovery

  In this case, if powerfail recovery support is included in the executive, COL will issue a SPRA$ directive. Upon powerfail recovery, COL will simply exit.

- Write errors

  If COL detects write errors, it will print a message and exit. Write errors will generally corrupt the logfile with invalid data.

- Automatic 8-hour termination.

COL Switches

Switch defaults --

```
     Output:  SY:TRACE.LOG/DENS:800/EX:100./IF:100./WI:10.
     Input:   SY:TRACE.MPB/BF:2:8/DL:0/DV:LBO/PR:GEN
```

Output Switches	Description
/AP	Append records to end of magtape. COL will space over all existing files on the magtape and append the records for this run starting at the current end-of-volume point.
/DENS:density	Set tape density to 800 or 1600 bpi.
/EX:n	Specifies disk file extent size in 256W blocks.
/IF:n	Specifies disk file initial size in 256W blocks.
/WI:n	Specifies number of retrieval pointers in window block.
Input Switches	
/BF:m:n	Number of buffers = m ; size of buffer = n 256W blocks.
/DL:n	Number of MINUTES COL delays before it starts data collection.
/DV:ddnn	Device selected for system metrics.
/PR:parnam	Partition selected for system metrics.
/TI:nn	Collect data only for tasks with TI of TTnn.

Example 1

```
>COL DB2:[3,300]CPUTEST/IF:1000/EX:200=DB1:[5,200]PARM1/DV:DB2:
```

logfile is DB2:[3,300]CPUTEST.LOG with initial size 1000 and extent size 200.

MPB file is DB1:[5,200]PARM1.MPB with system metrics specified for DB2.

Example 2

```
>COL MM1:/DENS:1600=DB1:[5,200]SAMPLE.MPB
```

logfile is on MM1: in 1600 BPI; MPB file is DB1:[5,200]SAMPLE.MPB.

Example 3

```
>COL =
```

logfile is SY:[current uic]TRACE.LOG; MPB file is SY:[current uic]TRACE.MPB.

NOTE: You cannot run COL with a null command line, i.e, >COL <CR>. The COL task expects to see explicit filespecs or a bare "=" indicating defaults on both input and output. Moreover, you cannot invoke COL with a command of the form:

```
                >RUN COL
```

since no filespecs can be specified with this form. One final thought in this regard: only one copy of COL can be active in the system at a time. If multiple invocations of COL are attempted at different terminals, all but the first will be rejected with an error message "MONITORING ACTIVITY IN PROGRESS".

Once COL is activated, it will run in the mode specified in the MPB but always for a maximum of 8 hours. This time limit is imposed because the KW11-P clock time will overflow in a little over 8 hours.

COL Messages

COL outputs two types of messages: error messages and status messages.

Full ASCII error message text will be printed only if the COLMSG task is installed. If the COLMSG task is not installed, COL will print a message of the form "COL ERROR XXXXXX", where XXXXXX is the error number. The error messages are listed by error number in the following list. Error messages are issued if an error condition occurs while COL is starting up and initializing the performance monitor. Error messages are output to the terminal of issue (TI:) for COL.

Status messages are issued while COL is running in steady state. There are status messages for both normal and error conditions. Status messages are issued to the console terminal (CO:). The COLMSG task does not have to be installed to get the status messages; as they are printed directly by the COL task.

COL Error Messages

000000 COLMSG -- XXXXXX (DSW) GET COMMAND LINE ERROR

Explanation: A "Get MCR Command Line" directive failure occurred.

User Action: Reenter COL command line using correct form.

NOTE: Illegal command forms are:

1. `>COL <CR>`
2. `>RUN $COL <CR>`

Legal command line forms are:

1. `>COL filespec/sw = filespec/sw`
2. `>COL filespec/sw = /sw`
3. `>COL =filespec/sw`
4. `>COL =`

000001 COLMSG -- COMMAND SYNTAX ERROR

Explanation: Entered incorrect syntax for command line.

User Action: Reenter command line with correct syntax.

000002 COLMSG -- MPB FILENAME SYNTAX ERROR

Explanation: Entered incorrect syntax for MPB filename.

User Action: Reenter MPB filename using correct syntax.

000003 COLMSG -- XXXXXX (FCS) MPB FILE ACCESS FAILURE

Explanation: COL failed to open MPB file or received a read error when attempting to read the MPB file. The code XXXXXX (FCS) indicates the FCS error code returned to COL.

User Action: Check that specified MPB file exists, the volume is mounted (if disk), or that the file protection does not prevent COL from reading the MPB.

000004 COLMSG -- XXXXXX (DSW) ASSIGN LUN FAILURE

Explanation: COL failed to assign a LUN to the logfile device.

User Action: Check that the device specified for logfile is in the system configuration.

000005 COLMSG -- ILLEGAL LOGFILE DEVICE TYPE

Explanation: The device specified for the logfile is not of the correct type. Valid logfile device types are: Directory structured (disk, DECTAPE), or Sequential (magtape).

User Action: Specify a device which is either directory structured or sequential.

000006 COLMSG -- LOG FILENAME SYNTAX ERROR

Explanation: Entered incorrect syntax for logfile.

User Action: Reenter logfile specification using correct syntax.

000007 COLMSG -- XXXXXX (FCS) LOGFILE ACCESS FAILURE

Explanation: COL failed to open the logfile. The code XXXXXX (FCS) is the FCS error returned to COL.

User Action: Check that the logfile volume is mounted and the necessary directory exists (if disk), or the volume is online (if magtape).

000010 COLMSG -- ILLEGAL BUFFER SPECIFIED

Explanation: The parameters specified by the /BF:n:m switch were not valid. Either the number of buffers (n) was out of range (<1 or >8) or the buffer size was out of range (>16).

User Action: Respecify a valid buffer count or size via the /BF switch.

000011 COLMSG -- XXXXXX (DSW) BUFFER PARTITION NOT IN SYSTEM

Explanation: The partition specified in the MPB is not in the system.

User Action: If the buffer partition is a static common, check that the static common partition exists and that the common is installed.

000012 COLMSG -- BUFFERS WILL NOT FIT INTO PARTITION

Explanation: The number of buffer blocks specified in the MPB (NUMBER OF BUFFERS * BUFFER SIZE IN 512 B BLOCKS) is greater than the number of 512 B blocks in the physical buffer partition.

User Action: Do a SMPGEN and specify the NUMBER OF BUFFERS and BUFFER SIZE IN 512 B BLOCKS such that the product of the two is less than the number of 512 B blocks in the physical buffer partition.

000013 COLMSG -- XXXXXX (DSW) CANNOT CREATE BUFFER PARTITION OR ADDRESS WINDOW

Explanation: COL received a DSW error when attempting to create the dynamic buffer partition specified in the MPB, or when attempting to create an address window to map the dynamic partition. The error XXXXXX (DSW) is the DSW error from the Create Region directive or from the Create Address Window directive.

User Action: Check that enough space exists in the GEN partition to contain a dynamic region with the size specified in the MPB.

If the error was from a Create Address Window directive, check the size of the COL task (look at the map in [11,2]COL.MAP) and verify that it is less than or equal to 4096 words. If COL is greater than 4096 words, it may have been taskbuilt using a SYSLIB.OLB with ANSI or big buffering FCS routines. COL must then be re-taskbuilt using a baseline SYSLIB.OLB which contains the non-ANSI FCS routines.

000014 COLMSG -- MONITORING ACTIVITY IN PROGRESS

Explanation: A multiple invocation of the COL task was attempted. Only one copy of COL can be active at a time.

User Action: Do not attempt to run more than one copy of COL at a time.

000015 COLMSG -- INVALID MPB FILE

Explanation: The MPB filespec specified in the COL command line is not the name of a valid MPB file.

User Action: Reenter a filespec for a valid MPB file.

000016 COLMSG -- INFORMATION RECORD ALLOCATION FAILURE

Explanation: Insufficient space exists in the buffer partition to contain all the Information Records.

User Action: Do a SMPGEN and specify enough blocks (NUMBER OF BUFFERS * BUFFER SIZE IN 512 B BLOCKS) to contain all the Information Records. The space required is roughly that needed to contain the TCBs, PCBs, DCBs, and UCBs in the system at the time COL is run.

000017 COLMSG -- KW11-P VECTOR OR CSR UNAVAILABLE

Explanation: Either the KW11-P vector is in use, or the KW11-P CSR is non-existent on the UNIBUS.

User Action: Check that the correct KW11-P vector and CSR were specified in the procedure @PMONBLD.CMD, that the vector is not used by another RSX-11M driver or task, and that the CSR is addressable on the UNIBUS.

000020 COLMSG -- CANNOT USE SYSTEM'S KW11-P as SPM-11M CLOCK

Explanation: The RSX-11M system clock is a KW11-P and an attempt was made to use it as the SPM-11M (performance monitor) clock.

User Action: Check that a separate KW11-P was installed for use by the SPM-11M software, and that the vector and CSR specified in the procedure @PMONBLD.CMD was for this clock.

000021 COLMSG -- POOL NODE ALLOCATION FAILURE

Explanation: COL could not allocate a RSX-11M pool node.

User Action: Re-run COL when sufficient pool is available.

000022 COLMSG -- BUFFER OVERFLOW

Explanation: COL failed to allocate space for a hook record because all buffers were full.

User Action: Re-run COL, using a larger buffer size specified either via SMPGEN.CMD or the /BF switch.

000023 COLMSG -- PARTITION SPECIFIED NOT IN SYSTEM

Explanation: The partition specified via the /PR switch is not configured in the system.

User Action: Respecify a partition configured in the system.

000024 COLMSG -- ILLEGAL DELAY VALUE SPECIFIED

Explanation: An illegal delay value has been specified via the /DL switch.

User Action: Re-specify a delay value in the correct range (0 to 28800).

COL Status Messages

COL -- PERFORMANCE MONITOR INITIALIZED

Explanation: COL has successfully initialized and is now ready to collect data.

User Action: None

COL -- PERFORMANCE MONITOR EXITING

Explanation: COL has stopped data collection and has cleaned up its databases. This status message is issued by COL immediately prior to exiting.

User Action: None

COL -- CANNOT MAP OUTPUT BUFFER

Explanation: COL received an error when attempting to map to the output buffer using the Map directive.

User Action: Check that the dynamic region used as the output buffer was successfully created.

COL -- LOGFILE WRITE ERROR

Explanation: COL received an FCS or QIO error when attempting to write to the logfile.

User Action: Check the integrity of the logfile. If repeated errors occur to the same logfile, rerun COL to create a new disk file or mount a fresh tape.

COL -- LOGFILE CLOSE ERROR

Explanation: COL received an error when attempting to close the logfile. COL will exit shortly after printing this message with the logfile improperly closed.

User Action: Because the logfile was improperly closed, it may be difficult to read the logfile and produce a performance report.

COL -- BUFFER OVERFLOW

Explanation: Hook records were written to the hook buffer faster than COL could write them to the logfile.

User Action: In general, the data collected during a measurement interval which overflowed is not usable by the MDR task to produce a performance report. An attempt should be made to rerun the measurement to obtain data without overflow. Prior to attempting another measurement run, a SMPGEN should be run with larger buffers specified than for the measurement run which overflowed.

3.2.3 Using TRM

The TRM task is used to manually signal the COL task to stop data collection, cleanup and exit. TRM will cause COL to exit, regardless of the mode in which COL was running: manual, autostop or repeat.

TRM is run by the command >RUN TRM.

3.2.4 Using MDR

MDR is used to read the logfile and produce a performance report. MDR is used in conjunction with DRS (sorting) task to produce the performance report. The general procedure is as follows:

-
  ```
  >RUN MDR
  MDR>LISTFILE/sw=LOGFILE/sw
  ```

  MDR will read the command line and execute pass 1. When pass 1 completes, MDR will print the following message and suspend:

  ```
  MDR -- YOU MUST NOW SORT THE INTERMEDIATE FILE BY
         USING DRS.TSK

  MDR --
    >DRS RESOURCE.SRT=RESOURCE.TMP/FO:V:156/KE:1.12/PR:T

  MDR -- AFTER THE SORT HAS COMPLETED, RESUME THIS TASK
         (...MDR)
  ```

NOTE: Rather than type the command line to DRS as shown above, you can simply invoke the command file @DRS.

- >@DRS sorts the intermediate file.

- >RES MDR

  MDR executes pass 2 and prints the performance report.

Certain MDR command switches restrict the input data which will be processed. If a logfile contains too much activity for MDR to reduce the data without running out of space, these switches (/AF, /BF, /-MD, /NT, /NU, /TS, /UC) may be used to reduce a logfile by successive runs of MDR, analyzing only parts of the workload in each run.

When using the switches /NT, /NU, /TS, or /UC, MDR will prompt for the task or device names after the command line is processed. The tasknames are specified as 1 to 6 RAD50 characters per entry. Only generic tasknames are used for /NT or /TS. The TI device names are entered as "DDnn:", with one entry per line.

MDR Switches

Switch defaults --

output: SY:REDUCE.RPT/SP

input: SY:TRACE.LOG/DM:2/LI:2/RW/SI:613/MD/SK:0/DE:800/-IC

Output Switches	Description
/SP /-SP	SPool output file
Input Switches	
/AF:PChigh:PClow	Only process records from AFter specified time
/BF:PChigh:PClow	Only process records from BeFore specified time
/DE:nnnn	Select input magtape DEnsity, 800 or 1600 (ignored unless /RW is also used)
/DM:nn	DuMp specified record types: 01 - buffer headers 02 - unknown record types 04 - system configuration info records 10 - task event hook records 20 - system metric records 40 - user information hook records
/IC /-IC	Replace the % resource column of the report with a count of incomplete intervals.
/LI:nn	Detail level of task workload metrics: 00 - no report produced 01 - system-wide summary only 02 - system and task summaries 03 - system, task, and TI: summaries 04 - all summaries and each complete task run 05 - all summaries and all task invocations

(continued on next page)

MDR Switches (continued)

Input Switches	Description
/MD /-MD	Report on tasks with missing data
/NT *	Do Not process hooks for listed Tasks
/NU *	Do Not process hooks for listed devices as TI:
/SC /-SC	Produce System Configuration section of report
/SI:nn	Selects the contents of Sampling Interval report by setting nn to the total for the desired items: 000 - no report produced 001 - data collection start/stop times 002 - % CPU usage 004 - % memory partition usage 010 - % device time busy 020 - % device space used 040 - % POOL used 100 - % CKPT space used 200 - clock time of metrics 400 - trailing blank line after the metrics
/SK:nnn	SKip nnn files when opening tape input file
/RW /-RW	ReWind tape input file before opening
/TS *	Process hook records only for listed TaSks
/UC *	Process hook records only for listed devices as TI:

* - the switches /NT and /TS are mutually exclusive.
 - the switches /NU and /UC are mutually exclusive.

MDR Messages

The MDR messages are issued to the TI device under which MDR runs. This section discusses the messages which MDR can issue.

```
MDR -- PLEASE ENTER TASKNAMES ONE AT A TIME FOLLOWING EACH
MDR -- WITH A <CR>.  TYPE JUST A <CR> AFTER THE LAST ONE IS
       ENTERED
MDR -- TSKNAM>
```

Explanation: This is the prompt for the generic tasknames for filtering by the /NT or /TS switch. The last line of this message is repeated for each line of input.

User Action: Enter generic task name(s).

```
MDR -- PLEASE ENTER TI DEVICES (DDNN:) ONE AT A TIME FOLLOWING
       EACH
MDR -- WITH A <CR>.  TYPE JUST A <CR> AFTER THE LAST ONE IS
       ENTERED
MDR -- TI DEVICE>
```

Explanation: This is the prompt for device names to be used by the /UC or /NU filter. The last line of this message is repeated for each line of input.

User Action: Enter full device name(s) as "DDnn:".

```
MDR -- YOU MUST NOW SORT THE INTERMEDIATE FILE BY USING DRS.TSK
MDR -- (E.G.) MCR>DRS
       RESOURCE.SRT=RESOURCE.TMP/FO:V:156/KE:1.12/PR:T
MDR -- AFTER THE SORT HAS COMPLETED, RESUME THIS TASK (TSKNAM)
```

Explanation: This message indicates that the first pass of MDR has completed. Manual intervention is now required to run the DRS task with the given command. The task name under which MDR is running is given in parenthesis in the last line of the message.

User Action: Run the DRS task (@DRS.CMD), and when it has completed, resume the MDR task.

```
MDR -- DSW:######  F.ERR:######  IOSB:######  ######  PC:######
       SP:######
MDR -- STK:###### ###### ...
```

Explanation: This is a status message which gives detail of the internal state of MDR at the time when an error is detected. This message is always followed by one of the messages discussed below, which describes the actual error condition.

User Action: See action specified for the message issued immediately following this status message.

MDR -- AN INCOMPLETE RESOURCE UTILIZATION REPORT WILL BE PRODUCED

Explanation: An error occurred which prevented the completion of pass II of MDR. Although a report will be produced, not all task-workload data has been summarized. A previous error message is issued giving the cause of this condition.

User Action: As specified for previous message.

MDR -- ATTEMPT TO PRINT NULL TWB

Explanation: An internal inconsistency occurred in the pass II processing of MDR.

User Action: * * * SPR !

MDR -- ATTEMPTING TO DEALLOCATE TWB NOT IN TSKHD

Explanation: An internal inconsistency occurred in the pass I processing of MDR.

User Action: * * * SPR !

MDR -- CHECKPOINT USAGE > 100%

Explanation: A Checkpoint metric indicates more than 100% usage of the checkpoint space. The metric is not reported.

User Action: Ensure logfile is not corrupted.

MDR -- CLOCK STOPPED OR HOOK RECORD OUT OF TIME SEQUENCE.

Explanation: The records in the logfile do not follow a strict time sequence. This condition invalidates the data in the logfile.

User Action: Ensure that the KW11-P clock is functioning. Re-collect the data.

MDR -- COMMAND READ ERROR

Explanation: An I/O error occurred while attempting to read the MDR command line.

User Action: Reenter the MDR command.

MDR -- COMMAND SYNTAX ERROR

Explanation: The command as entered has improper syntax.

User Action: Reenter the corrected MDR command.

MDR -- COMPLETION ERROR IN INPUT READ

Explanation: An I/O error occurred while reading the logfile. This condition will be treated as end-of-file and MDR will attempt to continue. This may be symptomatic of a corrupted logfile.

User Action: Check IOSB for error cause. Rerun MDR using another logfile if necessary.

MDR -- CONFLICTING SWITCHES /TS+/NT OR /UC+/NU

Explanation: The switches /TS and /NT were both specified, or the switches /UC and /NU were both specified. Only one of each pair of switches may be specified.

User Action: Reenter the command line without both of the conflicting switches.

MDR -- DEVICE USAGE > 100%

Explanation: MDR processed a system metric record which indicates greater than 100 percent of device utilization. No graphic output will be produced for the suspect record.

User Action: * * * SPR !

MDR -- DEVICE'S ACP NOT FOUND

Explanation: The system information records indicated that a device was mounted, yet no TCB for the ACP task was found. The device may have been dismounted concurrently with COL's scan of the system configuration. A generic ACP name will be used for the device's ACP functions.

User Action: None.

MDR -- DYNAMIC SPACE ALLOCATION FAILURE

Explanation: MDR was unable to sufficiently extend itself to provide space for initialization of pass I. This indicates a severe shortage of necessary dynamic space.

User Action: Ensure MDR is checkpointable. Ensure the partition where MDR runs has enough space for MDR to extend. Use SET /MAXEXT to allow sufficient extension. Use of smaller buffers by COL will lower the MDR space requirement.

MDR -- ERROR ATTACHING, POSITIONING OR SELECTING DENSITY OF MAGTAPE

Explanation: An I/O error occurred when attempting to open the logfile on magtape. The drive may have been mounted to MTAACP which is incorrect since MDR does QIOs directly to the tape device driver.

User Action: Correct the condition and rerun MDR.

MDR -- ERROR CLOSING INPUT FILE

Explanation: An I/O error occurred when closing the input logfile.

User Action: None.

MDR -- ERROR CLOSING REPORT FILE

Explanation: An I/O error occurred when closing the report file. This file will not be properly closed or spooled.

User Action: Correct the condition and manually close the file, or rerun MDR.

MDR -- ERROR CLOSING RESOURCE.TMP FILE

Explanation: An I/O error occurred at the end of pass I when closing the RESOURCE.TMP file. The DRS task will not be able to read this file until it is properly closed.

User Action: Correct the condition and manually close the file, or rerun MDR.

MDR -- ERROR DELETING RESOURCE.TMP FILE

Explanation: An I/O error occurred when MDR deleted its temporary file. The file is not properly closed, or deleted.

User Action: Manually unlock and delete this file.

MDR -- ERROR IN GTSK$S

Explanation: The directive failed preventing MDR from obtaining its taskname. The message requesting the sort to be run will not include MDR's taskname.

User Action: None.

MDR -- ERROR IN INPUT READ QIO

Explanation: An I/O error occurred while reading the logfile. This condition will be treated as end-of-file and MDR will attempt to continue. This may be symptomatic of a corrupted logfile.

User Action: Check DSW for error cause. Rerun MDR using another logfile if necessary.

MDR -- ERROR OPENING RESOURCE.SRT FILE

Explanation: An error was reported by FCS when MDR attempted to open the RESOURCE.SRT file for pass II. Pass II will not be able to run.

User Action: Correct the condition indicated by F.ERR and rerun MDR.

MDR -- ERROR OPENING RESOURCE.TMP FILE

Explanation: An error was reported by FCS when MDR attempted to open the RESOURCE.TMP file for write in pass I. Pass I will not be able to run.

User Action: Correct the condition indicated by F.ERR and rerun MDR.

MDR -- ERROR OUTPUTTING PRINT LINE

Explanation: An error was reported by FCS when MDR attempted to output a record to the report file. MDR will attempt to continue to produce the report. If this error is not caused by a transient condition the message will be repeated, and MDR will have to be aborted.

User Action: Correct the condition and rerun MDR.

MDR -- ERROR READING MPB

Explanation: An I/O error occurred when attempting to read the first block of the logfile.

User Action: Correct the condition and rerun MDR.

MDR -- ERROR READING RESOURCE.SRT FILE

Explanation: An I/O error occurred in pass II when attempting to read from the RESOURCE.SRT file. This will be treated as an end-of-file condition.

User Action: Correct the condition and rerun MDR.

MDR -- ERROR REOPENING RESOURCE.TMP FILE

Explanation: An I/O error occurred when attempting to re-open the RESOURCE.TMP file so it could be deleted. The RESOURCE.TMP file will not be deleted.

User Action: Delete the file manually.

MDR -- ERROR SKIPPING OVER MPB BUFFER

Explanation: An I/O error occurred when repositioning the tape logfile so the MPB could be reread once the buffer size is known.

User Action: Correct the condition and rerun MDR.

MDR -- ERROR WRITING RESOURCE.TMP FILE

Explanation: An I/O error was reported by FCS for a write operation during pass I.

User Action: Correct the condition and rerun MDR.

MDR -- EXPECTED RIB NOT FOUND

Explanation: An inconsistency occurred in MDR's internal dynamic data structures.

User Action: See Chapter 7, Section 7.3.

MDR -- EXPECTED RWB PREDECESSOR NOT FOUND IN OVERLAP

Explanation: An inconsistency occurred in MDR's internal dynamic data structures.

User Action: * * * SPR !

MDR -- EXPECTED RWB PREDECESSOR NOT FOUND IN MERGE

Explanation: An inconsistency occurred in MDR's internal dynamic data structures.

User Action: * * * SPR !

MDR -- IDLE TIME > 100%

Explanation: MDR processed a system metric record which indicates greater than 100 percent of CPU idle time. No graphic output will be produced for the suspect record.

User Action: See Chapter 7, Section 7.5.

MDR -- IMPROPER FORMAT OR NO SUCH DEVICE IN SYSTEM. PLEASE RETYPE.

Explanation: An improper response was entered to the prompt for TI filter devices.

User Action: Enter a corrected response.

MDR -- INPUT FILESPEC OR SWITCH SYNTAX ERROR

Explanation: An improper command line was used with MDR.

User Action: Reenter the command.

MDR -- INVALID DENSITY SELECTED, 800 OR 1600 ONLY

Explanation: A magtape density other than 800 or 1600 was specified.

User Action: Reenter the command with the correct density.

MDR -- INVALID DEVICE SPACE METRIC RECORDED

Explanation: A system metric record indicates more than 100% of device space used. The device space metric will not be reported. This metric is only available for devices mounted to F11ACP.

User Action: Ensure logfile not corrupted.

MDR -- INVALID INPUT DEVICE TYPE

Explanation: An attempt was made to reduce a logfile which is not on either a disk or tape volume. MDR can only use disk or tape for the input file.

User Action: Do not use MDR to read logfiles not residing on disk or tape.

MDR -- INVALID INPUT FILESPEC

Explanation: A syntax error occurred in the input filespec part of the MDR command.

User Action: Correct and reenter the command.

MDR -- INVALID LISTING LEVEL

Explanation: An invalid listing level was specified in the /LI switch. Acceptable values range from 0 to 5.

User Action: Correct and reenter the command.

MDR -- INVALID MPB FOUND BY ALC

Explanation: The logfile was found to contain improper data. This may be a result of using MDR to read a logfile produced by an incompatable COL version. This may also be the result of using MDR to read a file which was not produced by COL, or was corrupted.

User Action: Rerun MDR using a valid logfile.

MDR -- INVALID MPB FOUND BY INF

Explanation: The MPB did not pass validity testing when it was reread once the logfile buffer size was known. The logfile may have been corrupted while MDR was processing it.

User Action: * * * SPR !

MDR -- INVALID POOL METRIC RECORDED

Explanation: A Pool metric indicates more than 100% DSR usage. The metric is not reported.

User Action: Ensure logfile is not corrupted. Do not change DSR size while COL is running.

MDR -- INVALID RAD50 CHARACTER IN YOUR INPUT. PLEASE RETYPE.

Explanation: The response to the taskname prompt for filtered names was not a valid RSX-11M taskname. All letters should be in uppercase.

User Action: Correct and retype the response.

MDR -- INVALID TIME VALUES; /AF MUST BE EARLIER THAN /BF

Explanation: An attempt to use time filter values which overlapped causes this message. This condition will not allow any records to pass the filter for processing.

User Action: Correct and reenter the MDR command.

MDR -- MEMORY USAGE > 100%

Explanation: MDR processed a system metric record which indicates greater than 100 percent of partition utilization. No graphic output will be produced for the suspect record.

User Action: * * * SPR !

MDR -- MISSING INFO FOR DEVICE METRIC

Explanation: The Info record describing the system metric device was not found. Device system metrics will not be reported.

User Action: Do not load or unload device drivers while COL is active. Ensure logfile is not corrupted.

MDR -- MISSING I$SYS FOR POOL METRIC

Explanation: The DSR size was not found in the logfile. The Pool system metric is suppressed.

User Action: Ensure logfile is not corrupted.

MDR -- MISSING "=" AND INPUT FILESPEC

Explanation: The input command contains no "=" and no input filespec. Both input and output filespecs must be supplied or defaulted.

User Action: Enter a corrected MDR command including the "=".

MDR -- MORE THAN ONE INPUT FILESPEC NOT ALLOWED

Explanation: The MDR command contains more than one input filespec.

User Action: Enter a corrected MDR command with only one input filespec.

MDR -- MORE THAN ONE OUTPUT FILESPEC NOT ALLOWED

Explanation: The MDR command contains more than one output filespec.

User Action: Enter a corrected MDR command with only one output filespec.

MDR -- MPB FOUND AT LOCATION OTHER THAN FIRST RECORD OF FILE

Explanation: An MPB was read in the wrong position in the logfile. This is indicative of a corrupted logfile. MDR will bypass this MPB and attempt to continue to process the corrupted logfile. Reports produced from this logfile are suspect.

User Action: Use a different logfile for analysis.

MDR -- MPB NOT FOUND - NOT TRACE FILE

Explanation: The MDR input file did not begin with a MPB. Either the file was corrupted or it is not a logfile.

User Action: Use a different logfile for analysis.

MDR -- NO FOLLOWING PRIMARY RWB FOUND

Explanation: An internal inconsistency was detected in pass I.

User Action: * * * SPR !

MDR -- NO FPP OR BUILT /-FP

Explanation: MDR is unable to initialize for FPP ASTs. Either MDR was taskbuilt with the /-FP switch, or the processor where MDR is running has no FPP. This message cannot occur if the EIS version of MDR is used.

User Action: Rebuild MDR using the distributed .CMD file. Only run the FPP version of MDR on processors with an FP-11.

MDR -- OUTPUT FILESPEC OR SWITCH SYNTAX ERROR

Explanation: A syntax error was detected in the output filespec or an invalid output switch was specified.

User Action: Correct and reenter the MDR command.

MDR -- OVERFLOW BEFORE MPB - BAD TRACE FILE

Explanation: The logfile indicates buffer overflow occurred before the MPB was written. Either the logfile was corrupted, or the file was not produced by COL.

User Action: Use another file for MDR input.

MDR -- OVERFLOW IN FIRST PASS SUMMARIZATION

Explanation: A floating overflow occurred in pass I. Incorrect data will be reported for the particular task and summaries which this data represents.

User Action: Use the resulting report with caution. Use filtering to limit the amount of data processed to prevent this condition.

MDR -- OVERFLOW IN SECOND PASS SUMMARIZATION

Explanation: A floating overflow occurred in pass II. Incorrect data will be reported for the particular task and summaries which this data represents.

User Action: Use the resulting report with caution. Use filtering to limit the amount of data processed to prevent this condition.

MDR -- PARTITION RIB NOT FOUND FOR S$MEM

Explanation: A system metric record was read for a partition which was not described by the system configuration information records. The graphic output of memory utilization will be suppressed.

User Action: None.

MDR -- PRINT LINE TOO LONG

Explanation: MDR generated a report file record longer than 132 bytes. The line is truncated by the logic which writes to the file.

User Action: * * * SPR !

MDR -- RIB LIST LINKAGE ERROR

Explanation: An inconsistency was detected in MDR's internal representation of the system configuration.

User Action: * * * SPR !

MDR -- RWB WITHOUT TWB FOUND IN RESOURCE.TMP

Explanation: A record in the RESOURCE.SRT file was encountered in an illogical position. The record is disgarded.

User Action: * * * SPR !

MDR -- SUMMARIZING FROM NULL TWB

Explanation: An inconsistency was detected in the pass II data structures.

User Action: * * * SPR !

MDR -- SUMMARIZING INTO NULL TWB

Explanation: An inconsistency was detected in the pass II data structures.

User Action: * * * SPR !

MDR -- SVTK FAILURE

Explanation: The RSX executive rejected MDR's attempt to specify an SST vector.

User Action: Correct the condition indicated by the DSW and rerun MDR.

MDR -- SYSTEM + IDLE TIME > 100%

Explanation: A system metric record was read which indicates the total of kernel and idle time is greater than the elapsed time. The graphic output for the record is suppressed.

User Action: See Chapter 7, Section 7.5.

MDR -- SYSTEM TIME > 100%

Explanation: A system metric record was read which indicates more kernel time than elapsed time. The graphic output for the record is suppressed.

User Action: See Chapter 7, Section 7.5.

MDR -- TI: READ ERROR

Explanation: An I/O error occurred when reading filter values from the user.

User Action: Rerun MDR.

MDR -- TRAP ###

Explanation: A floating point or conversion error occurred. The octal number in this message indicates the precise error cause.

User Action: * * * SPR !

MDR -- TRUNCATED INPUT FILE RECORD

Explanation: The end of a buffer was reached before all the records in that buffer were processed. This indicates a corrupted logfile.

User Action: Use a different logfile for analysis.

MDR -- TWB/RWB/RIB DYNAMIC SPACE ALLOCATION FAILURE

Explanation: MDR was unable to get needed virtual space for a dynamic structure in pass I. However sufficent dynamic storage was obtained to begin pass I.

User Action: Ensure MDR is checkpointable. Ensure the partition where MDR runs has enough space for MDR to extend. Use SET /MAXEXT to allow sufficient extension. Use of smaller buffers by COL will lower the MDR space requirement. Use of filtering of MDR input will also lower this space requirement. Use overlaid MDR.

MDR -- UNABLE TO ALLOCATE RESOURCE.SRT BUFFER

Explanation: MDR was unable to get needed virtual space for a dynamic structure in pass II.

User Action: Ensure MDR is checkpointable. Ensure the partition where MDR runs has enough space for MDR to extend. Use SET /MAXEXT to allow sufficient extension. Use of filtering of MDR input will lower this space requirement. Use overlaid MDR.

MDR -- UNABLE TO ALLOCATE RIB

Explanation: MDR was unable to sufficiently extend itself to provide space for initialization of pass I. This indicates a severe shortage of necessary dynamic space.

User Action: Ensure MDR is checkpointable. Ensure the partition where MDR runs has enough space for MDR to extend. Use SET /MAXEXT to allow sufficient extension. Use of smaller buffers by COL will lower the MDR space requirement.

MDR -- UNABLE TO OPEN INPUT FILE

Explanation: An I/O error occurred when MDR attempted to open the logfile.

User Action: Correct the condition and rerun MDR.

MDR -- UNABLE TO OPEN OUTPUT FILE

Explanation: An I/O error occurred when MDR attempted to open the report file.

User Action: Correct the condition indicated by F.ERR and rerun MDR.

MDR -- UNKNOWN INFO TYPE RECORD FOUND

Explanation: A corrupted logfile, or a file not produced by COL is being used as MDR input.

User Action: Use a different file for MDR input.

MDR -- UNKNOWN RECORD TYPE

Explanation: A corrupted logfile, or a file not produced by COL is being used as MDR input.

User Action: Use a different file for MDR input.

MDR -- VALIDATION ERROR

Explanation: MDR is being used to process a logfile after the service period has expired.

User Action: Contact your local Digital office to extend the service period.

MDR -- WILDCARD NOT ALLOWED IN FILESPEC

Explanation: The MDR command contains a wildcard.

User Action: Reenter the command without any wildcards.

MDR -- WRONG FOLLOWING PRIMARY RWB FOUND

Explanation: An inconsistency was detected in MDR's pass I dynamic data.

User Action: * * * SPR !

3.2.5 Using CLF

CLF is used to copy logfiles produced by COL. The three modes of CLF operation are:

- Buffer Transfer Mode,

  which copies a logfile from either disk or tape, to disk or tape, maintaining the logfile format which MDR uses for input.

- FCS Record Transfer Mode,

  which copies a logfile from either disk or tape, producing an output file with variable length FCS records.

- Read-only Mode,

  which reads the logfile, computes overflow and buffer counts, but does not create a new file.

The general procedure for using CLF is:

```
>RUN CLF
CLF> [OUTFILE/sw =] LOGFILE/sw
```

Note that the output filespec and "=" are optional. If these are not specified, CLF will operate in read-only mode, producing no output file.

When FCS mode is specified (by the /FCS switch), the records have the format specified in appendices A through D. The first word of each record is the record ID which uniquely identifies the record type. If overflows occur within the data stream, a special "overflow record" is placed in the output file at the point where records were lost due to overflow. The format of this (length = 4. bytes) record is:

600	0	600(8) = overflow record ID
###	2	count of records lost by overflow at this point

When the /FCS switch is used, the default output filespec is changed to TRACE.FCS.

After CLF has completed reading the input file, the run statistics will be reported to CLF's TI:.

CLF Switches

Switch defaults --

```
     output:  SY:TRACE.LOG/-AP/DE:800/-FCS

     input:  SY:TRACE.LOG/DE:800/RW/SK:0
```

Output Switches	Description
/AP /-AP	APpend file to the end of magtape. CLF will space over all existing files on the magtape and write the new file starting at the current end-of-volume point.
/DE:nnnn	Selects output magtape DEnsity, 800 or 1600
/FCS /-FCS	Selects FCS formatted output file
Input Switches	
/DE:nnnn	Selects input magtape DEnsity, 800 or 1600 (ignored unless /RW is also used)
/RW /-RW	ReWind tape input file before opening
/SK:nnn	SKip nnn files when opening tape input file

CLF Messages

The CLF messages are issued to the TI device under which CLF is run. This section discusses the messages which CLF can issue.

```
CLF -- #### BUFFERS READ
CLF -- #### RECORDS READ
CLF -- #### BUFFERS WITH OVERFLOW
CLF -- #### RECORDS LOST BY OVERFLOW
```

Explanation: This statistics message is output at the completion of each run of CLF, indicating the statistics about the input file.

User Action: None.

```
CLF -- DSW:###### F.ERR:###### IOSB:###### ###### PC:######
       SP:######
CLF -- STK:###### ###### ...
```

Explanation: This is a status message which gives detail of the internal state of CLF at the time when an error is detected. This message is always followed by one of the messages discussed below, which describes the actual error condition.

User Action: See action specified for the message issued immediately after this status message.

CLF -- COMMAND READ ERROR

Explanation: An I/O error occurred while reading the command line.

User Action: Reenter the command.

CLF -- COMMAND SYNTAX ERROR

Explanation: The command has improper syntax.

User Action: Reenter the corrected command.

CLF -- COMPLETION ERROR IN INPUT READ

Explanation: an I/O error occurred while reading the input file. This condition, which may be symptomatic of a corrupted logfile, will be treated as end-of-file.

User Action: Check F.ERR and IOSB for error cause, or use another logfile.

CLF -- ERROR ASSIGNING INPUT DEVICE

Explanation: CLF was unable to assign a LUN to the input file.

User Action: Check DSW and F.ERR for cause. Ensure the filespec syntax is correct.

CLF -- ERROR ASSIGNING OUTPUT DEVICE

Explanation: CLF was unable to assign a LUN to the output file.

User Action: Check DSW and F.ERR for cause. Ensure the filespec syntax is correct.

CLF -- ERROR ATTACHING, POSITIONING OR SELECTING DENSITY OF MAGTAPE

Explanation: An I/O error occurred when attempting to open a file on magtape.

User Action: Ensure the drive is not mounted if logfile format is used for that file. Ensure the output drive is selectable, and not write locked. Correct the error indicated by DSW or IOSB.

CLF -- ERROR CLOSING INPUT FILE

Explanation; An I/O error occurred when closing the input file.

User Action: None.

CLF -- ERROR CLOSING OUTPUT FILE

Explanation: An I/O error occurred when truncating or closing the output file.

User Action: Correct the condition and rerun CLF, or manually unlock the output file.

CLF -- ERROR IN INPUT READ QIO

Explanation: An I/O error occurred while reading the input file. This condition, which may be symptomatic of a corrupted logfile, will be treated as end-of-file.

User Action: Check DSW for cause. Rerun using another logfile if necessary.

CLF -- ERROR READING MPB

Explanation: An I/O error occurred when reading the first block of the logfile.

User Action: Correct the condition and rerun.

CLF -- ERROR WRITING TO OUTPUT FILE

Explanation: An I/O error occurred when writing to the output file.

User Action: Correct the condition and rerun.

CLF -- INPUT FILESPEC OR SWITCH SYNTAX ERROR

Explanation: A syntax error was detected in the input filespec.

User Action: Reenter the corrected command.

CLF -- INVALID DENSITY SELECTED, 800 OR 1600 ONLY

Explanation: A magtape density other than 800 or 1600 was specified.

User Action: Reenter the command with a correct density.

CLF -- INVALID INPUT DEVICE TYPE

Explanation: An attempt was made to read a file not on disk or tape. Disk and tape are the only acceptable media for logfiles.

User Action: Do not use CLF to read files not on disk or tape.

CLF -- INVALID MPB FOUND

Explanation: The first block of the input file does not pass certain validity tests which ensure the file is a logfile. This may be caused by a corrupted or obsolete logfile.

User Action: Use a different logfile.

CLF -- INVALID OUTPUT DEVICE TYPE

Explanation: An attempt was made to write in logfile format to a device other than disk or tape.

User Action: Only write logfile format to disk or tape.

CLF -- MORE THAN ONE INPUT FILESPEC NOT ALLOWED

Explanation: The command has improper syntax.

User Action: Reenter the corrected command.

CLF -- MORE THAN ONE OUTPUT FILESPEC NOT ALLOWED

Explanation: The command has improper syntax.

User Action: Reenter the corrected command.

CLF -- MPB NOT FOUND - NOT TRACE FILE

Explanation: The first block of the input file does not pass certain validity tests which ensure the file is a logfile. This may be caused by a corrupted or obsolete logfile.

User Action: Use a different logfile.

CLF -- OUTPUT FILESPEC OR SWITCH SYNTAX ERROR

Explanation: A syntax error was detected in the input filespec.

User Action: Reenter the corrected command.

CLF -- OVERFLOW BEFORE MPB - BAD TRACE FILE

Explanation: The first block of the input file does not pass certain validity tests which ensure the file is a logfile. This may be caused by a corrupted or obsolete logfile.

User Action: Use a different logfile.

CLF -- TRUNCATED INPUT FILE RECORD

Explanation: The end of a buffer was reached before all records in that buffer were processed. This indicates a corrupted logfile.

User Action: Use a different logfile.

CLF -- UNABLE TO OPEN INPUT FILE

Explanation: An I/O error occurred while opening the input file.

User Action: Correct the condition and rerun.

CLF -- UNABLE TO OPEN OUTPUT FILE

Explanation: An I/O error occurred while opening the output file.

User Action: Correct the condition and rerun.

CLF -- UNKNOWN RECORD TYPE

Explanation: An unknown record ID was found in the file.

User Action: Use a different logfile.

CLF -- WILDCARD NOT ALLOWED IN FILESPEC

Explanation: A wildcard was found in the command.

User Action: Reenter the command without wildcards.

3.2.6 Writing User Information to the Logfile

The QPDRV driver provides a way for a user task to write user information to the same logfile that COL currently has open. A task does this by issuing a QIO to QP:, with up to six words of user information specified in the QIO parameter words.

The QPDRV driver is similar to the null driver NLDRV. It is an infinite sink for writes and an infinite source of EOFs for reads. QPDRV is compatible with FCS and RMS QIO functions. All of the following QIO functions are defined as legal and control for QPDRV: IO.KIL, IO.RLB, IO.WLB, IO.ATT, IO.DET, IO.FNA, IO.ACR, IO.ACW, IO.ACE, IO.DAC, IO.RVB and IO.WVB.

QPDRV contains a hook point defined as H$USER. When a QIO is issued to QP:, a hook record with hook ID H$USER is written. The H$USER record contains the six QIO parameter words specified by the user. See Appendix A for a complete specification of the hook record.

To use QPDRV, you must load it by the command:

```
        >LOA QP:
```

QPDRV is built via the command file PMONBLD.CMD. The image and symbol table files for QPDRV are contained in [1,54].

Example:

```
        ALUN$C  1,QP,0
        QIOW$C  IO.WLB,1,1,,,, <100,200,300,400,500,600>
```

will result in a H$USER record with the QIO parameters 100, 200, 300, 400, 500 and 600 written to the logfile.

To dump a stream of H$USER records in the performance report, you run MDR and use the /DM:40 switch. If you only want the H$USER records printed, the suggested command syntax is:

```
        MDR> output=input/DM:40/LI:0/SI:0
```

Usage note: If you enable device type "OTHER" in SMPGEN, then for each QIO to QP:, three hook records are written: H$QDRV, H$IODN and H$USER. However, if you disable device type "OTHER" in SMPGEN, then for each QIO to QP: only the H$USER hook record is written.

Chapter 4
DATA COLLECTION AND REDUCTION PROCEDURES

COL and MDR are complementary tools which can be used to collect large amounts of performance data and interpret the data to a great degree of precision. Both COL and MDR have a number of parameters and switches which provide flexibility in both collecting and interpreting the data. In the following sections we describe how to utilize the parameters of COL and MDR to collect the best possible set of data to analyze your computer system.

4.1 DATA COLLECTION PROCEDURE

There are a number of questions you should answer before running COL to collect data:

- Length and type of measurement interval?
- Resource types enabled for data collection?
- Device classes enabled for data collection?
- Tasks enabled for data collection?
- System metrics enabled?

4.1.1 Measurement Period

The measurement period is defined as the time during which COL is collecting data. Typically this time is one continuous time interval; however, it may consist of a number of time intervals piecewise added together.

Under most conditions, you should select a continuous measurement period. This is done by running COL either in MANUAL mode or AUTOSTOP mode.

MANUAL mode should be used when:

- There is an operator available to terminate COL.
- You estimate the logfile space available is sufficient to contain all the data collected for the measurement period.

AUTOSTOP mode should be used when:

- An operator is not available to terminate COL.
- You want to run COL for a fixed amount of time (AUTOSTOP on TIME).
- You want to set a limit on the amount of logfile space used (AUTOSTOP on BUFFER).

Under some conditions, particularly if the system is under heavy activity for long periods of time, COL will collect too much data given the available logfile space. To reduce the logfile space used, you can run COL in REPEAT mode. However, you should do so carefully. The chief concern when running COL in REPEAT mode is that "incomplete intervals" may result if a task or operation is outstanding exactly when COL starts or stops data collection. For example, if you start COL precisely when a task is active and has an I/O operation queued to the disk, an incomplete interval will be flagged by MDR for both the task invocation and for the I/O operation. This condition does not invalidate other information in the performance report, it just "leaves out" some information which potentially may be valuable. You can minimize the number of incomplete intervals by minimizing the number of times COL enables or disables data collection for a measurement period.

Let's say you want to collect data during a normal 8-hour daytime shift and you have only one 2400 foot magtape available for the logfile. However, you estimate that if COL runs all day, about two 2400 foot tapes will be needed. What you need to do is cut the amount of data collected in half. Consider the following alternative measurement schemes:

- Sample time = 1 minute; Time between samples = 1 minute.

- Sample time = 30 minutes; Time between samples = 30 minutes.

- Sample time = 2 hours; Time between samples = 2 hours.

Scheme 1 will usually result in the highest number of incompletes, yet it probably will not "miss" any important system activity. Scheme 3 will probably result in the lowest number of incompletes, yet it is vulnerable to missing important intervals of system activity. Scheme 2 will probably result in a low number of incompletes, yet it has a good chance of capturing all periods of important system activity.

What we're observing here is an old problem in communications systems: How fast do you sample a signal to be able to reconstruct the signal from the sampled data? In our case, the problem is: how fast do you sample the workload in order to reconstruct it?

In communication systems, you sample the signal at twice the highest frequency in the signal. In computer systems, it would be reasonable to sample the workload at twice the highest frequency of variation in the workload. Thus if you observe the workload to fluctuate up and down about four times per day, then 8 samples per day (every 30 minutes) would reduce the amount of data and yet collect sufficient data to reconstruct the workload.

4.1.2 Resource Types

Normally you want to collect data for all resource types (CPU, memory, I/O). If the amount of data collected is too large, you can select one resource type per measurement period and run a number of measurement periods.

The hooks are grouped according to resource type. To select a particular resource type, you select the corresponding hook group.

4.1.3 Device Classes

The device classes are: Directory, Sequential, Terminal, and Other. If you want to measure the resource contention for a device, select it for data collection. Normally the disks are shared devices and should be selected. If the line printer is spooled and attached to the despooler task you may not need to collect usage data for it. Similarly, terminals are normally singly used by one task at a time; the measurement results may not be useful from a resource contention point of view.

4.1.4 Tasks

The tasks selected for data collection are the definition of the workload that you want to measure. For example, if you want to analyze the application workload, select all the tasks in that workload and deselect the system service tasks such as F11ACP, MCR..., and QMG.... On the other hand, if you want to measure the system resources that a particular F11ACP is consuming, enable only F11ACP and disable all other tasks.

4.1.5 System Metrics

Normally system metrics should be enabled with a sample interval of 60 seconds. If you want to save logfile space and report paper, longer sample intervals may be used.

4.2 DATA REDUCTION PROCEDURE

Once you have a logfile on disk or tape, you are ready to read the logfile with MDR and reduce the data to a usable performance report. A guide to reading the performance report is given in Chapter 5.

The data reduction procedure should consist of a number of steps methodically performed. A general guideline is to first obtain a number of overviews of the system activity during the entire measurement period and then print detailed analyses of the subset measurement intervals of interest. For example, if during an 8 hour shift there were two intervals of peak activity, from 9 to 11 a.m. and from 2 to 4 p.m., you should first run MDR with /LI:0 to get a system metrics scan of the entire 8 hour shift, and then run MDR with /AF and /BF to get a report of only the peak morning activity and again with /AF and /BF to get only the peak afternoon activity. The general procedure is as follows:

- Run MDR with /LI:0 for entire logfile.

  Identify the intervals of interest and record their endpoints of time using the KW11-P clock timestamps printed above the system metrics.

- Rerun MDR once for each subset measurement interval using /AF and /BF with the /LI:2 switch.

This will give both system and task summaries. Identify the largest consumers of resources and record their tasknames.

- Rerun MDR with /LI:4 or /LI:5 and /TS for the tasks desired.

  Observe a detailed analysis of the particular tasks which consumed the resources and the times when these resources were consumed.

Chapter 5
READING THE PERFORMANCE REPORT

5.1 PERFORMANCE REPORT DATA

When reading this section, refer to Appendix F, which contains a sample performance report.

The performance report consists of four sections:

- Measurement Parameters
- System Configuration
- System Metrics
- Task Metrics

5.1.1 Measurement Parameters

The measurement parameters are read from the Monitor Parameter Block (MPB) which is the first block of the logfile. The MDR task extracts the information from it and prints the measurement parameters on the first page of the performance report. The items printed are:

- Label.
- Measurement mode.
- Buffer type, partition, number and size.
- Resources selected for System Metrics.
- TI filter device (if selected).
- Tasks selected for data collection.

- Devices selected for data collection.
- Hooks enabled.
- MDR command line

5.1.2 System Configuration

The system configuration information is contained in the Information Records and directly follows the MPB in the logfile. If the /SC switch is specified, the MDR task will extract the information from these records and print it to the logfile. The information printed is:

- The Pool (DSR) size in bytes and the executive feature mask.
- A list of the partitions in the system when COL was activated.
- A list of the controllers and units in the system when COL was activated.
- A list of the tasks installed in the system when COL was activated.
- A list of the active checkpoint files and their sizes when COL was activated.

5.1.3 System Metrics

The system metrics are system-wide measurements of CPU usage, pool usage, the usage of a selected partition, checkpoint file usage, and the usage of a selected device. Measurements are made for each sampling interval and the results written to the logfile. The length of the sampling interval is selected in the SMPGEN.CMD procedure. The system metrics are:

- CPU Usage

  For each sampling interval, counters are kept for the total times spent in kernel and idle states. The remainder of the sampling interval time is assumed to be user state. To depict CPU usage, a histogram is printed on the performance report. Each 1% of time spent by the CPU in user mode is represented by a single letter "U". Each 1% of time spent by the CPU in kernel mode is represented by a single letter "K".

The user time is left justified and grows from left to right; the kernel time is right justified and grows from right to left. If any blank spaces exist in the line, they represent CPU idle time.

- Pool Usage

  At the end of each sampling interval, COL scans the RSX-11M system pool and measures the percentage of pool in use. The percentage of pool used is represented by a histogram, where each "P" represents 1% of pool usage.

- Partition Usage

  The user selects a partition to be monitored via the COL switch /PR:parnam. At the end of each sampling interval, COL scans the partition data structures for the selected partition to measure the percentage of the memory in use in the partition. The percentage of memory used in the partition is represented by a histogram, where each "M" represents 1% of partition usage.

- Checkpoint File Usage

  When COL is activated, it scans the checkpoint file list and records a list of all active checkpoint files and their sizes.

  At the end of each sampling interval, COL scans the checkpoint file list and measures the percentage of all active checkpoint file space in use. If there is more than one checkpoint file, COL sums the space in all checkpoint files to compute the percentage. The percentage of checkpoint file space used is represented by a histogram, where each "C" represents 1% of checkpoint file space usage.

- Device Usage

  The user selects a device to be monitored via the COL switch /DV:ddnn. For each sampling interval, COL tallies all the I/O request times for the selected device to get the percentage of time that the I/O device was in use during the sampling interval. The percentage of time the device was used is represented by a histogram, where each "D" represents 1% of device usage time for the sampling interval.

In addition, if the device is a Files-11 disk and is mounted native, the number of disk blocks used on the volume is computed at the end of the sampling interval. The percentage of disk space used is represented by a histogram where each "B" represents 1% of the total volume space used.

For each sampling interval, the histograms for CPU, pool, memory, checkpoint file space, and device are clustered together into six successive lines on the report.

The end of each sampling interval is timestamped and printed in two formats: KW11-P time and RSX-11M system time.

The KW11-P time is printed as two octal numbers: the left number is the high order part and the right number is the low order part. The corresponding RSX-11M system time is printed as HRS:MIN:SEC.TENTHS.

The KW11-P timestamp can be used to:

- Correlate an individual task report page (which has the same form of timestamp) with a particular system metric line.

- Select a particular measurement interval of interest by rerunning MDR using the /AF and /BF switches.

The totals for CPU kernel time, CPU idle time and device busy time are given for each measurement interval.

The last item in the system metrics section is the "Total Data Reduction Period". This is the total time for which data was reduced. If the report was generated using the /AF and /BF switches, only the data reduced between /AF and /BF is counted.

5.1.4 Task Metrics

<u>Levels of Task Summarization</u>

The task metrics describe the usage of the resources in the system on a task-by-task basis. There are a number of levels of summarization available in the performance report, which enables the user to choose between a very detailed report or a shorter summarized report. The levels of summarization and their corresponding switches are as follows:

- Individual Task Invocations (/LI:4 or /LI:5)

  An individual task invocation is defined as a single run of a task, that is, the time between an executive Run Task event and an executive Exit Task event (the exact placement of the hooks for these events is specified in Appendix A). A complete set of task metrics will be printed for each task invocation. If most or all tasks in the system are enabled for data collection, a report containing individual task invocations may be quite long.

- Task by TI Summary (/LI:3)

  In this level of summary, the task metrics for tasks with the same generic taskname (see Chapter 2, section 2.2.2) and the same TI are grouped together when the statistics are computed. The task metrics then reveal the measurements summed across all tasks with the same generic taskname from the same TI.

- Task Summary (/LI:2)

  In this level of summary, the task metrics for tasks with the same generic taskname (see Chapter 2, section 2.2.2) are grouped together when the statistics are computed. The task metrics then reveal the measurements summed across all tasks with the same generic taskname.

- System-Wide Summary (/LI:1)

  In this level of summary, the task metrics for all tasks in the system that were enabled for data collection are grouped together when the statistics are computed. The task metrics then reveal the measurements summed across all tasks in the system.

Task Metric Data

The following data is presented in the performance report for all levels of task summarization.

- Generic TASK

  The generic taskname (see Chapter 2, Section 2.2.2) is printed on the top of the page to identify the task for which the data applies.

- Elapsed Times

  The elapsed time is the time from a Run Task event to an Exit Task event.

- Number of Task Invocations

  The Number of Task Invocations is the number of times that a Run Task event occurred for the task. If the COL task is started while a task is in progress, the Run Task event will not be recorded in the logfile for that task. When the task exits, an Exit Task event will be recorded to the logfile. When MDR reads an Exit Task event without a corresponding Run Task event, it reports an "incomplete" invocation in the Number of Task Invocations line.

- Memory Times

  There are three memory times:

  - In (memory)

  - Out (memory) - includes time waiting for memory on initial task load and all time checkpointed (waiting and stopped).

  - Wait (memory) - includes time waiting for memory on initial task load and time checkpointed (waiting only).

- Memory Size in K Words

  Size of the task region.

- Space-Time Product

  The space-time product is the amount of memory that a task uses integrated over the time that the memory was used. Each increment of memory used (in K words) is multiplied by the time (in SECs) that the increment was used, and all products are summed to get the space-time product. If a task changes its size via an Extend Task directive, the corresponding change in memory usage is computed.

- ACP Counts

The ACP Counts are tallied across all ACPs in the system. They are not broken down by separate ACP.

There are seven categories of ACP functions printed on the performance report. They are:

- Access/Deaccess

  These functions are: Access File for Read, Access File for Read/Write, Access File for Read/Write/ Extend, Deaccess File, Connect Logical Link, and Disconnect Logical Link, and Close Out LUN.

- Create/Delete

  These functions are: Create File, Delete File and Truncate File.

- Read/Write

  These functions are: Read Virtual Block, Write Virtual Block, Receive Message and Transmit Message.

- Directory

  These functions are: Find File Name in Directory, Enter File Name in Directory and Remove File Name from Directory.

- File Attribute

  These functions are: Read File Attributes and Write File Attributes.

- Extend

  This is the Extend File function.

- Other

  These functions are all ACP functions other than those listed above, such as: Unlock Block, User Magtape Control Function and Network Control Function.

- Memory Residency

  The Memory Residency count is incremented each time a task successfully allocates memory for an initial task load or for a checkpoint read.

- CPU Timeslices

  A CPU Timeslice is the time between a Load Task Context event and a Save Task Context event, minus the idle time accrued during that period. Note that if a task issues a directive and the processor switches to kernel mode, the corresponding time in kernel mode is charged to the active task's CPU timeslice. Also, if an interrupt occurs while a task's context is active, the corresponding kernel mode time is charged to the active task's CPU timeslice.

- Initial Loads

  Initial Loads is the number of Initial Task Load events for the task. The USAGE times indicate the time it took the loader task to load the task; the WAIT times indicate how long a task's load request was in the loader task's queue.

- Checkpoint R+W

  Checkpoint R+W is the number of Checkpoint Read and Checkpoint Write events for the task. The USAGE times indicate the time it took the loader task to checkpoint (read or write) the task. (Note: the USAGE times do not indicate how long a task was checkpointed. This information appears under "Memory Times" on the task metric page.)

- DDnn Ovly Loads

  DDnn Ovly Loads is the number of overlay read (IO.LOV) QIOS the task issued to device DDnn.

- DDnn QIOs

  DDnn QIOs is the number of logical QIOs a task issued to device DDnn. Please note the following caveats:

  - DDnn Ovly Loads is included in this count, since IO.LOV is a Logical block QIO.

- Task IO.RVB and IO.WVB (virtual) QIOs which are successfully and entirely mapped by the executive or F11ACP to IO.RLB and IO.WLB (Logical) QIOs are included in this count.

- Task IO.RVB and IO.WVB (virtual) QIOs which are sent to F11ACP, and which are performed by F11ACP on behalf of the task (partially mapped, multiple transfers), are not included in this count. This logical block I/O is charged to F11ACP.

- The IO.ATT, IO.DET and IO.KIL QIOs are not included in this count.

USAGE Times

The USAGE times indicate how long a task used a resource.

WAIT Times

The WAIT times indicate how long a task was waiting for the use of a resource.

SERVICE Times

The SERVICE time is the USAGE time plus WAIT time which occurred for each resource request. Note that the minimum SERVICE time is not necessarily equal to the minimum USAGE time plus minimum WAIT time, since the minimum USAGE time may have occurred for a different resource request than the resource request which had the minimum WAIT time. The same applies to the maximum.

ARRIVALS PER SECOND

The mean task rate is the number of resource requests issued by a task divided by the task's elapsed time. It tells, from a task's point of view, the rate of resource requests issued by a task. The mean system rate is the number of resource requests issued by a task divided by the data reduction period. It tells, from a resource's point of view, the rate of resource requests issued by a task.

Resource Metrics

For each resource used by a task, the following metrics are printed on the report:

- Count

  The number of times a task used a resource.

- Total Time

  The total time during which a task used the resource (equal to the sum of all resource usage intervals).

- % Resident Time

  % Resident Time is (Total Time/Resident Time) * 100%.

- Minimum

  The smallest interval during which a task used a resource.

- Mean

  The arithmetic average of all resource usage intervals for a task.

- Maximum

  The largest interval during which a task used a resource.

- Coefficient of Variation

  The coefficient of variation is the standard deviation of the intervals divided by the mean interval.

- % Resource

  % resource is the total time a task used the resource divided by the total time of the Data Reduction Period (total resource time available).

- Incomplete

  If the logfile does not contain a "complete" set of events needed to define an interval, MDR will tally the interval as an "incomplete". For example, if MDR reads a I/O Done event from the logfile for a certain task and device and there is no corresponding Get I/O Packet event for the task and device, MDR will regard the I/O interval as an incomplete. Incomplete intervals can occur if COL starts data collection while an operation (interval) is already in progress.

The only data recorded for incomplete intervals is the count; no timing information is computed since the information is not precise.

Incomplete intervals are printed in lieu of the % resource metric if the /IC switch is specified.

Chapter 6
RESOURCES USED BY SPM-11M

The purpose of this section is to give an estimate of the resources used by the COL and MDR tasks.

6.1 RESOURCES USED BY COL

- CPU

  When a task metric hook is encountered, from 50 to 75 instructions are executed at system level (PR=0, $STKDP=0) before returning to the main flow of the executive.

- System Pool

  When the COL task is inactive, about 115 words of executive space are used. This is accounted for by the hook points (3 words per hook), 4 words added to system common, 16 words added to the system stack and a dummy hook return routine added to SYSXT.

  When the COL task is activated, approximately 215 more words of system pool are used.

- Physical Memory

  The COL task occupies about 4K words of physical memory. The COL buffers (either in a static common or dynamic region) can vary from 0.5K words to 32K words.

- I/O Requests

  The COL task issues a FCS WRITE$ to write a buffer to disk; it issues a QIO$ to write a buffer to magtape. The rate of I/O requests generated by COL depends on the system activity that COL is tracing.

6.2 RESOURCES USED BY MDR

- Memory

  The code and static data of MDR requires varying amounts of memory depending upon which version is built. See the TASK IMAGE SIZE in MDR.MAP. Additional space is required for I/O buffers and dynamic structures which are manipulated within MDR's virtual address space. MDR will automatically extend its virtual space as needed to accommodate these structures. Typically MDR will require an additional 8KW for all the dynamic storage.

- Pool

  All MDR I/O is singly buffered, and synchronous, minimizing the use of pool.

- I/O

  Each pass of MDR reads a single file and writes to the report file. During pass I the RESOURCE.TMP file is also being written. Terminal I/O is used for command input, and error reporting. The DRS task will have up to 7 files concurrently opened for I/O.

Chapter 7
USAGE NOTES
SPM-11M

7.1 TASKBUILDING COL

COL must be taskbuilt using the baseline library [1,1]SYSLIB.OLB (which does not contain FCS routines with ANSI or big buffering).

If COL is taskbuilt with the ANSI or big buffering FCS routines, its size may be greater than 4096 words. The result will be that when COL is run, it will not be able to create an address window (the error received will be COLMSG -- CANNOT CREATE BUFFER PARTITION OR ADDRESS WINDOW).

7.2 COL BUFFERS

When COL starts up it copies all the TCBs, PCBs, DCBs and UCBs to the logfile. There must be enough buffer space to store all of these data structures simultaneously, otherwise COL will print the error message COLMSG -- INFORMATION RECORD ALLOCATION FAILURE.

Before COL is left to run unattended for long periods of time, COL should be run for 5 to 10 minutes under a typical heavy or peak system workload and the console terminal should be monitored for the occurrence of overflow errors, i.e., COLMSG -- BUFFER OVERFLOW. If buffer overflows occur, the buffer size for COL can be increased by using the /BF switch (see Chapter 3, section 3.2.2).

Another cause of COL overflows may be small disk file extent sizes. When COL attempts to write a buffer to the logfile, F11ACP may have to do an extend file operation before the actual data write can be done. The time it takes F11ACP to extend a file may be long, and COL may overflow as a result of this. To remedy this situation, two switches have been built into COL:

- Initial File Size /IF
- Extend File Size /EX

If an estimate can be made of the maximum logfile size before COL is run, it is best to preallocate all the logfile space via the /IF switch. In this case COL will not need F11ACP to do any extend file operations during its run and this may prevent overflows from occurring.

7.3 STATE OF SYSTEM AT COL STARTUP TIME

When COL starts up, it copies the TCBs, PCBs, DCBs and UCBs in RSX-11M to the logfile. The MDR task later refers to these data structures to derive the necessary context for printing a performance report.

To assure that MDR will have all the context it needs to print the report, you should insure that the following conditions are satisfied <u>before</u> starting up COL:

- All tasks installed

  If you want to disable a task for data collection, that task must be installed when COL starts up. All tasks dynamically installed after COL starts <u>will always be enabled for data collection</u>. This means that all multiple copies of a multiuser task will always be enabled for data collection.

- All partitions set up

  If you want memory metrics kept for a partition, the partition must be set up (via SET /MAIN=XYZPAR:m:n:type) before starting COL.

- All drivers loaded

  If you want the I/O metrics kept for a device-unit, the driver managing that device-unit must be loaded before starting COL.

- All devices mounted

  If you want the correct ACP taskname for a mounted volume to be printed on the performance report, the volume must be mounted before starting COL. Otherwise, MDR will print a "pseudo" taskname for the ACP.

Example: If the device DK1: is mounted to the F11ACP task after data collection is in progress for COL, MDR will print the pseudo name DK1***ACP on the report.

- All checkpoint files allocated

  If you want the system metrics for checkpoint file usage, be sure to allocate all checkpoint files with the ACS command before starting COL.

7.4 TASK BUILDING MDR

MDR will run on processors with or without hardware FPP (floating point processor) support. If your processor does not have hardware FPP, it must minimally have hardware EIS instructions.

You must taskbuild MDR to reflect the type of hardware (FPP or EIS) on your processor. You can build MDR either overlaid or non-overlaid with both FPP and EIS versions. To build MDR, invoke the build command file:

```
>@MDRTKB
```

This command file will prompt you to build any of the four available versions:

1. FPP support, overlaid
2. FPP support, non-overlaid
3. EIS support, overlaid
4. EIS support, non-overlaid

You can change certain MDR defaults by editing the taskbuild command file produced by MDRTKB.CMD. The MDRTKB.CMD procedure will prompt you for changing the defaults. The defaults which can be changed are:

- Number of lines per page in REDUCE.RPT file
- Default values for /LI switch
- Default values for /SI switch
- Default magtape logfile density (/DE)
- Default value for /DM switch

7.5 USE WITH LAYERED PRODUCTS

SMP-11M inserts hooks into the RSX-11M executive at standard places where the executive does allocations and deallocations of resources.

SPM-11M measures kernel CPU time by hooking the standard interrupt save ($INTSV, $INTSE, $INTSC) and system exit ($DIRXT) points.

If any layered product or user privileged component uses nonstandard techniques for resource allocation or interrupt entry, the SPM-11M measurements will be inaccurate.

For example, DECNET uses a nonstandard interrupt save routine, which means the kernel CPU time measurement will be inaccurate.

Appendix A
HOOK RECORD DEFINITIONS

NOTE: All record offsets in this appendix are expressed in octal bytes.

All record lengths are expressed in decimal bytes.

1.0 H$CDRP - CALL DIRECTIVE PROCESSOR

1.1 LOCATION

```
DRDSP.MAC
        After   $EMTRP::DIRSV$
        After   $DPLM2::MOV      (SP)+,R3
        After   50$:    MOV      R5,R2
        Follow          MOV      $HEADR,R4
```

1.2 DATA COLLECTION

1.2.1 Storage Dependencies -

$TKTCB (Current Task TCB)
$DICSV (Directive ID Code)

1.2.2 Record Format -

Hook I.D	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
DIC	14

1.2.3 Record Length -

14. bytes.

2.0 H$RTSK - REQUEST TASK TO RUN

2.1 LOCATION

```
REQSB.MAC
       After   $TSKRP::
       After   1$:
       Follow          MOV      R2,T.UCB(R0)
```

2.2 DATA COLLECTION

2.2.1 Storage Dependencies -

R0=TCB address

2.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
Request UIC	14

2.2.3 Record Size -

14. bytes.

3.0 H$XTSK - TASK EXIT

3.1 LOCATION

```
DREIF.MAC
        After   $CEXIT::
        After   134$:
        Follow  .ENDC      ;P$$OFF
```

3.2 DATA COLLECTION

3.2.1 Storage Dependencies -

$TKTCB=TCB address

3.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
Idletime	14

3.2.3 Record Length -

16. bytes.

4.0 H$SCTX - SAVE CONTEXT OF TASK

4.1 LOCATION

```
SYSXT.MAC
        After   RESCH:    CLR   $RQSCH
        Follow  73$:      MOV   H.GARD(R3),R2
```

4.2 DATA COLLECTION

4.2.1 Storage Dependencies -

$TKTCB (Current Task TCB)

4.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
Idletime	14

4.2.3 Record Length -

16. bytes.

5.0 H$LCTX - LOAD CONTEXT OF TASK

5.1 LOCATION

```
SYSXT.MAC
        After   RESCH:    CLR   $RQSCH
        After   120$:     MOV   H.GARD(R2),R0
        Follow            .ENDC
```

5.2 DATA COLLECTION

5.2.1 Storage Dependencies -

$TKTCB (TCB address of new task)

5.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
Idletime	14

5.2.3 Record Length -

16. bytes.

6.0 H$USTP - UNSTOP TASK

6.1 LOCATION

```
REQSB.MAC
   After     $EXRQN::BIT   #T2.STP*2!T2.STP,T.ST2(RO)
   Follow            BIC   #T2.STP*2!T2.STP,T.ST2(RO)
```

6.2 DATA COLLECTION

6.2.1 Storage Dependencies -

RO=TCB address

6.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12

6.2.3 Record Length -

12. bytes.

7.0 H$QAST - QUEUE AST TO TASK

7.1 LOCATION

```
REQSB.MAC
          After   $QASTT::ADD       #T.ASTL,RO
          Follow              CAll  $SETCR
```

7.2 DATA COLLECTION

7.2.1 Storage Dependencies -

RO=TCB address

7.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12

7.2.3 Record Length -

12. bytes.

8.0 H$EXTK - EXTEND TASK

8.1 LOCATION

```
DREXP.MAC
        After   $DREXP::TSTB    T.IOC(R5)
        Follow  71$:      MOV   R1,P.SIZE(R0)
```

8.2 DATA COLLECTION

8.2.1 Storage Dependencies -

$TKTCB=TCB address
R0=PCB address

8.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
Main PCB Address	14
Task Size	16

8.2.3 Record Length -

16. bytes.

9.0 H$QTRP - QUEUE TASK REQUEST FOR PARTITION

9.1 LOCATION

```
REQSB.MAC
        After   $TSKRP::
        After   7$:
        Follow          MOV   R1,R0
```

9.2 DATA COLLECTION

9.2.1 Storage Dependencies -

R0=TCB address

9.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12

9.2.3 Record Length -

12. bytes.

10.0 H$QLDR - QUEUE TASK LOAD REQUEST TO LOADER QUEUE

10.1 LOCATION

```
REQSB.MAC
       After   $NXTSK::SAVNR
       After   25$:
       Follow            BISB  -(R0),P.BUSY+1(R5)
```

10.2 DATA COLLECTION

10.2.1 Storage Dependencies -

R1=TCB address

10.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12

10.2.3 Record Length -

12. bytes.

11.0 H$QCHW - QUEUE CHECKPOINT WRITE REQUEST TO LOADER QUEUE

11.1 LOCATION

```
REQSB.MAC
       After   $CHKPT::
       After   50$:
       Follow              BIS   #TS.CKP,T.STAT(R1)
```

11.2 DATA COLLECTION

11.2.1 Storage Dependencies -

R1=TCB address

11.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
T.ST2	14

11.2.3 Record Format -

14. bytes.

12.0 H$GLDR - GET REQUEST FROM LOADER QUEUE

12.1 LOCATION

```
LOADR.MAC
       After   $LOADR::CAll      $SWSTK,40$
       Follow  10$:         MOV  R1,LDRTK
```

12.2 DATA COLLECTION

12.2.1 Storage Dependencies -

R1=TCB address

12.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
T.LDV	14
P.MAIN	16
P.SIZE	20
T.STAT	22

12.2.3 Record Format -

20. bytes.

13.0 H$FLDR - FINISH LOADER REQUEST

13.1 LOCATION

```
LOADR.MAC
       After   $LOADR::CAll    $SWSTK,40$
       Follow  84$:       CAll $SWSTK,$LOADR
```

13.2 DATA COLLECTION

13.2.1 Storage Dependencies -

R5=TCB address

13.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12

13.2.3 Record Length -

12. bytes.

14.0 H$CREG - CREATE REGION

14.1 LOCATION

```
DRREG.MAC
        After   $DRCRR::
        After   74$:
        Follow  BIS          #RS.CRR,R.GSTS(R3)
```

14.2 DATA COLLECTION

14.2.1 Storage Dependencies -

$TKTCB=TCB address
R4=Region's PCB address

14.2.2 Record Format -

Hook ID	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
Main Par. PCB Address	14
Region's PCB Address	16
P.SIZE	20
Region name (RAD50)	22

14.2.3 Record Length -

22. bytes.

15.0 H$DREG - DELETE REGION

15.1 LOCATION

```
DRREG.MAC
        After   $DETRG::MOV     A.PCB(R5),R0
        Follow  40$:    MOV     P.SUB(R0),P.SUB(R1)
```

14.2 DATA COLLECTION

15.2.1 Storage Dependencies -

$TKTCB=TCB address
R0=Region's PCB Address

15.2.2 Record Format -

Hook ID	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
Region PCB Address	14

15.2.3 Record Length -

14. bytes.

16.0 H$QDRV - QUEUE I/O PACKET TO DRIVER

16.1 LOCATION

```
DRQIO.MAC
       After   $DRQIO::
       Follow  $DRQRQ::MOV      U.SCB(R5),R4
```

16.2 DATA COLLECTION

16.2.1 Storage Dependencies -

R1=IRP address
R5=UCB address

16.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
UCB Address	14
Function code	16
LUN	20
IRP Address	22

16.2.3 Record Length -

20. bytes.

17.0 H$GPKT - DRIVER GETS AN I/O PACKET

17.1 LOCATION

```
IOSUB.MAC
       After   $GTPKT::
       Follow  128$:       CLC
```

17.2 DATA COLLECTION

17.2.1 Storage Dependencies -

R1=IRP address
R5=UCB address

17.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
IRP Address	14

17.2.3 Record Length -

14. bytes.

18.0 H$IODN - I/O DONE PROCESSING

18.1 LOCATION

```
IOSUB.MAC
       Follow  $IODON::MOV     U.SCB(R5),R4
```

18.2 DATA COLLECTION

18.2.1 Storage Dependencies -

R4= SCB Address

18.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
IRP Address	14

18.2.3 Record Length -

14. bytes.

19.0 H$IOFN - COMPLETE I/O POST PROCESSING

19.1 LOCATION

```
IOSUB.MAC
       Follow  $IOFIN::
```

19.2 DATA COLLECTION

19.2.1 Storage Dependencies -

R3=IRP address

19.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
IRP Address	14

19.2.3 Record Length -

14. bytes.

20.0 H$QAC1 - QUEUE I/O PACKET TO ACP

20.1 LOCATION

```
DRQIO.MAC
       After   $DRQIO::
       After   FCXIT:
       Follow            MOV  U.ACP(R5),R0
```

20.2 DATA COLLECTION

20.2.1 Storage Dependencies -

R1=IRP address

20.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
UCB Address	14
Function Code	16
LUN	20
IRP Address	22

20.2.3 Record Length -

20. bytes.

21.0 H$QAC2 - QUEUE I/O PACKET TO ACP

21.1 LOCATION

```
IOSUB.MAC
        After   $GTPKT::
        Follow  95$:
```

21.2 DATA COLLECTION

21.2.1 Storage Dependencies -

R1=IRP address

21.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
UCB Address	14
Function Code	16
LUN	20
IRP Address	22

21.2.3 Record Length -

20. bytes.

22.0 H$GACP - ACP GET PACKET

22.1 LOCATION

```
DISPAT.MAC (F11ACP)
       After   .START::
       Follow          MOV     $HEADR,R5
```

22.2 DATA COLLECTION

22.2.1 Storage Dependencies -

R1=IRP Address

22.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
IRP Address	14

22.2.3 Record Length -

14. bytes.

23.0 H$DRSV - DIRECTIVE SAVE

23.1 LOCATION

```
SYSXT.MAC
       After   $DIRSV::
       Follow                MOV   SP,@$HEADR
```

23.2 DATA COLLECTION

23.2.1 Storage Dependencies -

None

23.2.2 Routine -

Timestamps system entrance time $BSTIM.

23.2.3 Record Format -

None

24.0 H$INSV - INTERRUPT SAVE

24.1 LOCATION

```
SYSXT.MAC
       After   $INTSV::
       Follow                MOV   SP,@$HEADR
```

24.2 DATA COLLECTION

24.2.1 Storage Dependencies -

None

24.2.2 Routine -

Timestamps system entrance time $BSTIM.

24.2.3 Record Format -

None

25.0 H$INS2 - INTERRUPT SAVE

25.1 LOCATION

```
SYSXT.MAC
       After   $INTSC::
       Follow              MOV  SP,@$HEADR
```

25.2 DATA COLLECTION

25.2.1 Storage Dependencies -

None

25.2.2 Routine -

Timestamps system entrance time $BSTIM.

25.2.3 Record Format -

None

26.0 H$SYXT - SYSTEM EXITS TO USER TASK

26.1 LOCATION

```
SYSXT.MAC
       After   $DIRXT::
       Follow  5$:
```

26.1.1 Storage Dependencies -

None

26.1.2 Routine -

Copies current clock time, subtracts system entrance time ($BSTIM), and adds the resulting delta time to the total system time $TSTIM.

26.1.3 Record Format -

None

27.0 H$EIDL - ENTER EXECUTIVE IDLE LOOP

27.1 LOCATION

```
SYSXT.MAC
       After   RESCH:
       After   30$:
       Folow             BNE      10$
```

27.2 DATA COLLECTION

27.2.1 Storage Dependencies -

None

27.2.2 Routine -

Copies current clock time, subtracts the system entrance time ($BSTIM), and adds the resulting delta time to the total system time $TSTIM. Timestamps idle entrance time $BIDLE.

27.2.3 Record Format -

None

28.0 H$XID1 - EXIT EXECUTIVE IDLE LOOP

28.1 LOCATION

```
SYSXT.MAC
       After   RESCH:
       Follow  46$:     CLRB    (R1)
```

28.2 DATA COLLECTION

28.2.1 Storage Dependencies -

None

28.2.2 Routine -

Copies current clock time, subtracts the idle loop entrance time ($BIDLE), and adds the resulting delta time to the total idle time $IDLE. Timestamps system entrance time $BSTIM.

28.2.3 Record Format -

None

29.0 H$XID2 - EXIT EXECUTIVE IDLE LOOP

29.1 LOCATION

```
SYSXT.MAC
       After   $FORK0::
       Follow          BCC     $INTXT
```

29.2 DATA COLLECTION

29.2.1 Storage Dependencies -

None

30.0 H$USER - USER INFORMATION HOOK

30.1 LOCATION

QPDRV.MAC
Follow QPINI:

30.2 DATA COLLECTION

30.2.1 Storage Dependencies -

R1=IRP address
R5=UCB address

30.2.2 Record Format -

Hook I.D.	0
Timestamp	2
Taskname (RAD50)	6
TI UCB Address	12
Function Code	14
LUN	16
QIO Parameter 1	20
QIO Parameter 2	22
QIO Parameter 3	24
QIO Parameter 4	26
QIO Parameter 5	30
QIO Parameter 6	32

30.2.3 Record Length -

28. bytes.

Appendix B
MPB DEFINITIONS

Monitor Parameter Block

The Monitor Parameter Block (MPB) is the data structure which contains the measurement parameters. These measurement parameters are defined by invoking the procedure @SMPGEN.CMD, which produces:

- o The binary MPB file (filename.MPB)
- o The source MPB file (filename.PAR)
- o The MPB saved answer file (filename.CMD)

The COL task reads the MPB and initializes its databases from the MPB.

The MPB has the following format and contents:

Parameter	Offset	Size(wds)	Values
MPB ID word (I$MPB)	M.PID	1	I$MPB
Measurement label	M.LABL	32.	

Buffer Type	M.BTYP	1	DYNAMIC=0
			STATIC=1

Buffer Count	M.BCNT	1	
Buffer Size (bytes)	M.BSIZ	1	
Buffer Size (64b blk)	M.BS64	1	

Parameter	Offset	Size(wds)	Values
Buffer Partition Name	M.BPAR	2	
SYS Partition Name	M.BPNM	2	
Buffer Partition Addr.	M.BPAD	1	
Buffer Partition Size	M.BTSZ	1	
Buffer Partition Flags	M.BFLG	1	

Measurement Mode	M.MODE	1	MANUAL=0
			AUTOSTOP=1
			REPEAT=2

Sample Count (Buffers)	M.SCNT	1	
Sample Interval Mag.	M.SINT	1	
Sample Interval Units	M.SUNT	1	
Repeat Count	M.RCNT	1	
Repeat Interval Mag.	M.RINT	1	
Repeat Interval Units	M.RUNT	1	
Hook Bit Table	M.HTBL	4	
I/O Selection Word	M.IOS	1	
Other Devices Sel. Word	M.OTHR	1	
Device Class Mask	M.CW1M	1	
Device Selection Mask	M.CW1S	1	
Sys. Metrics Samp. Int.	M.SAMP	1	
Sys. Metrics Partition	M.PART	2	
Sys. Metrics Device	M.DEVC	2	
TI Filter Device	M.TIDV	1	
TI Filter Device Num.	M.TINM	1	
MPB Validation Word	M.VALD	1	P$$VER
Task Selection Mask	M.TKS	1	
Task Selection Table	M.TTBL	128.	

Appendix C
INFORMATION RECORD DEFINITIONS

Information Record

The Information Record provides system context data which can be correlated with the Hook Records to identify tasks, partitions, and I/O device-units.

In addition, there are Information Records which capture essential data when COL starts and stops data collection.

I$SYS - System Common Information

I$SYS	0
Initial Pool Size	2
$FMASK	4
Sys. Met. Dev. UCB Address	6

I$TCB - Task Control Block

I$TCB	0
I$TCB Record Size	2
TCB Address	4
TCB Contents • • •	6

I$PCB - Partition Control Block

I$PCB	0
I$PCB Record Size	2
PCB Address	4
PCB Contents • • •	6

I$DCB - Device Control Block

I$DCB	0
I$DCB Record Size	2
DCB Address	4
DCB Contents . . .	6

I$UCB - Unit Control Block

I$UCB	0
I$UCB Record Size	2
UCB Address	4
UCB Contents . . .	6

I$CKP - Checkpoint File Information

I$CKP	0
CKP. File Device UCB Addr.	2
CKP. File Size	4

I$TMR - Start Data Collection

I$TMR	0
System Time (Year)	2
System Time (Month)	4
System Time (Day)	6
System Time (Hour)	10
System Time (Minute)	12
System Time (Second)	14
System Time (Tick)	16
Ticks Per Second	20
High KW11-P Time	22
Low KW11-P Time	24

I$STM - Stop Data Collection

I$STM	0
System Time (Year)	2
System Time (Month)	4
System Time (Day)	6
System Time (Hour)	10
System Time (Minute)	12
System Time (Second)	14
System Time (Tick)	16
Ticks Per Second	20
High KW11-P Time	22
Low KW11-P Time	24
CPU Kernel Time (High)	26
CPU Kernel Time (Low)	30
CPU Idle Time (High)	32
CPU Idle Time (Low)	34
Device Usage Time (High)	36
Device Usage Time (Low)	40

Appendix D
SYSTEM METRIC RECORD DEFINITIONS

System Metrics Record

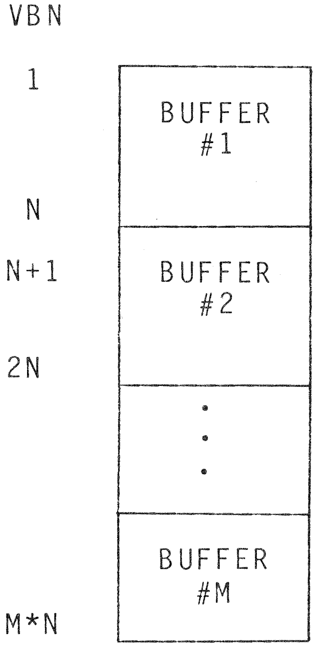
The System Metrics Record contains data which represents the system-wide usage of the CPU, pool, a selected memory partition, checkpoint file space, and a selected device.

S$MET	0
High KW11-P Time	2
Low KW11-P Time	4
CPU Kernel Time (High)	6
CPU Kernel Time (Low)	10
CPU Idle Time (High)	12
CPU Idle Time (Low)	14
# Free Pool Nodes	16
Total Free Pool Space (bytes)	20
Smallest Pool Node (bytes)	22
Largest Pool Node (bytes)	24
Main Partition PCB Address	26
Number of Subpartitions	30
Total Space Used (32W blks)	32
Total CKP.File Space	34
Total CKP.File Space Used	36
UCB Address of Device	40
Device Usage Time (High)	42
Device Usage Time (Low)	44
Free Disk Blocks (High)	46
Free Disk Blocks (Low)	50

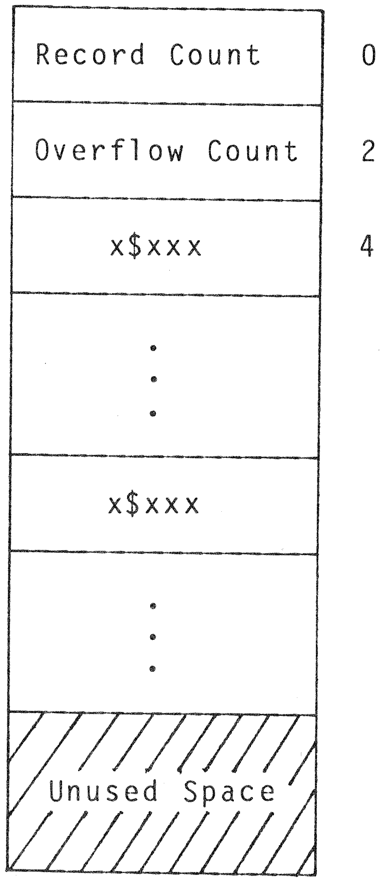
Appendix E
LOGFILE FORMAT

1.0 DISK LOGFILE FORMAT

A logfile on disk is a Files-11 file with the following format:



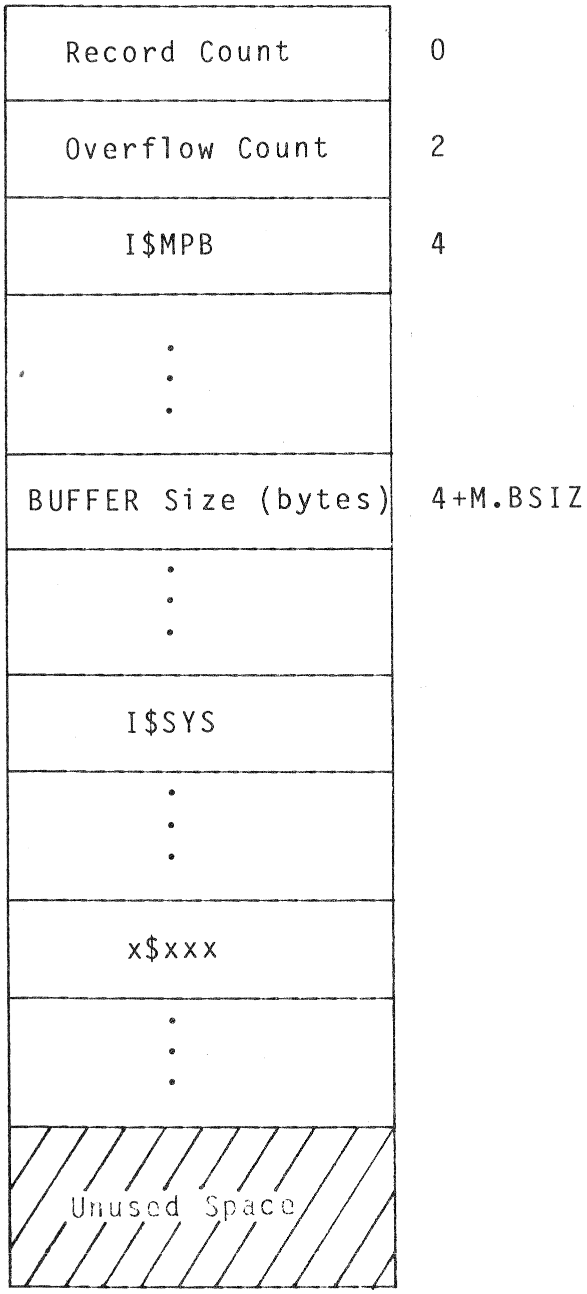
Each BUFFER has the following general format (byte offsets):



x$xxx is I$, H$ or S$ record ID. See appendices A, B, C and D.

All BUFFERS are the same length. The BUFFER size in 512B blocks is derived from the BUFFER size in bytes which is stored in the MPB in BUFFER #1:

BUFFER #1 Format (byte offsets)

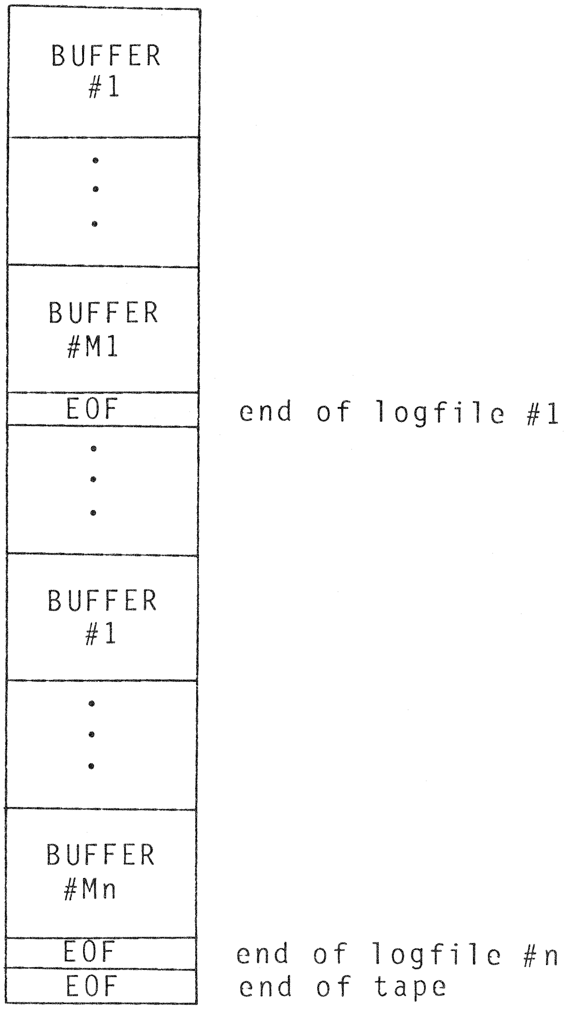


Note:
I$MPB is the 1st record;
I$SYS is the 2nd record;
x$xxx is I$, H$ or S$ record ID. See appendices A, B, C and D.

$$\text{BUFFER size (blocks)} = \frac{\text{BUFFER size (bytes)}}{512.}$$

2.0 MAGTAPE LOGFILE FORMAT

A magtape can contain one or more logfiles. Each BUFFER is terminated by an IRG. Each logfile is terminated by a tape mark (EOF). The magtape is terminated by a double tape mark (EOF EOF). Each BUFFER has the same format as for disk.



3.0 FCS LOGFILE FORMAT

The CLF utility task can be used to convert a logfile from the formats specified in section 1.0 and 2.0 to a file containing a stream of variable length FCS records. Each FCS record contains one record of type H$xxxx, I$xxxx or S$xxxx. The first word of each record contains the record ID, H$xxxx, I$xxxx or S$xxxx. The record formats are specified in Appendices A, B, C and D.

See Chapter 3, Section 3.2.5 for using CLF.

Appendix F
EXAMPLE PERFORMANCE REPORT

```
RSX-11M Performance Monitor Data Reduction V01.000         18-JUN-81 14:51:07         Page 1         Monitor Collection Parameters
                                          NEW VERSION 8-JUN-81

                     P E R F O R M A N C E    M O N I T O R    D A T A    C O L L E C T I O N    P A R A M E T E R S

Mode: MANUAL

Buffering in a DYNAMIC region called TRCPAR in Partition GEN    , of 3 Buffers with 8 Blocks per Buffer

System-wide metrics recorded every 60 seconds for the CPU, Partition GEN    , and Device LB0:

Task Workload DEVICES Selected:  DISKS TAPES

Task Workload Data Collection Enabled for All EXCEPT the following Task(s):
  F11ACP  ...MCR  MCR...  QMG...  TTP21   ...INS  ...RMD

the following HOOK POINTS are ENABLED:
(Directives)
(Run/Exit)      RTSK  XTSK
(CPU Usage)     SCTX  LCTX
(Memory)        USTP  QAST  EXTK  QTRP  CREG  DREG
(Task Loader)   QLDR  QCHW  GLDR  FLDR
(In/Output)     QDRV  GPKT  GACP  IODN  IOFN  QAC1  QAC2
(System/Idle)   DRSV  INSV  INS2  SYXT  EIDL  XID1  XID2

MDR command line>SAM2/-SP=DB:[11,2]SAM2/SI:777/TS
```

```
RSX-11M Performance Monitor Data Reduction V01.000         18-JUN-81 14:51:07         Page 2         Sampling Intervals
                                  NEW VERSION 8-JUN-81

Data Collection Started 15-JUN-81 09:33:58.0 000002 020246

               0         +        20         +        40         +        60         +        80         +       100  (% utilization)
ending at 000135:132212 (09:34:58.0):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU                                                            KKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space
  DB0: Busy    DDDDDDDDDDDDDDDDDDDDDDD
  DB0: Blks    BBBBBBBBBBBBBBB

ending at 000271:046075 (09:35:58.1):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU               KKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space
  DB0: Busy    DDDDDDDDDDDDDDDDDDDD
  DB0: Blks    BBBBBBBBBBBBBBB

ending at 000424:161144 (09:36:58.1):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU                               KKKKKKKKKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space   CCCCCCCC
  DB0: Busy    DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
  DB0: Blks    BBBBBBBBBBBBBBB

ending at 000560:074273 (09:37:58.1):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUKKKKKKKKKKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space   CCCCCC
  DB0: Busy    DDDDDDDDDDDDD
  DB0: Blks    BBBBBBBBBBBBBBB

ending at 000714:007727 (09:38:58.1):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUKKKKKKKKKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space   CCCCCC
  DB0: Busy    DDDDDDDDDDDDD
  DB0: Blks    BBBBBBBBBBBBBBB

ending at 001047:124250 (09:39:58.2):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUKKKKKKKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space   CCCCCCCC
  DB0: Busy    DDDDDDD
  DB0: Blks    BBBBBBBBBBBBBBB

ending at 001203:040673 (09:40:58.2):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUKKKKKKKKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
```

```
RSX-11M Performance Monitor Data Reduction V01.000         18-JUN-81 14:51:07         Page 3         Sampling Intervals
                                  NEW VERSION 8-JUN-81
  CKPT Space   C
  DBO: Busy    DDDD
  DBO: Blks    BBBBBBBBBBBBBBB

ending at 001336:155710 (09:41:58.2):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU                                                        KKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space   C
  DBO: Busy    DDDDDDDDDDDDDDDDDDDDD
  DBO: Blks    BBBBBBBBBBBBBBB

ending at 001472:072406 (09:42:58.2):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUU                                                 KKKKKKKKKKKKKKKKKKKKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space   C
  DBO: Busy    DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
  DBO: Blks    BBBBBBBBBBBBBBBB

ending at 001626:007407 (09:43:58.3):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU              KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space   C
  DBO: Busy    DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
  DBO: Blks    BBBBBBBBBBBBBBBB

ending at 001761:124306 (09:44:58.3):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU               KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space   C
  DBO: Busy    DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
  DBO: Blks    BBBBBBBBBBBBBBBB

ending at 002115:042047 (09:45:58.3):
  CPU  Usage   UUUUUUUUUUUUUUUUUUUUUUUUUUUUU                                         KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space   C
  DBO: Busy    DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
  DBO: Blks    BBBBBBBBBBBBBBB

ending at 002250:160750 (09:46:58.4):
  CPU  Usage   UUUUUU                                                                                            KKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space   C
  DBO: Busy    D
  DBO: Blks    BBBBBBBBBBBBBBB

ending at 002404:100126 (09:47:58.4):
  CPU  Usage   UUUUUUU                                                                                        KKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
```

```
RSX-11M Performance Monitor Data Reduction V01.000         18-JUN-81 14:51:07         Page 4          Sampling Intervals
                                  NEW VERSION 8-JUN-81

  CKPT Space   C
  DB0: Busy    DDDDDDDDD
  DB0: Blks    BBBBBBBBBBBBBBB

ending at 002540:017256 (09:48:58.5):
  CPU  Usage   UUUUUUUUUUUUUUUUUUU                                                                   KKKKKKKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
  CKPT Space
  DB0: Busy    DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
  DB0: Blks    BBBBBBBBBBBBBBB

ending at 002673:136265 (09:49:58.5):
  CPU  Usage   UUUUUUUUUUUUUUU                                                                         KKKKKKKKKKKKK
  POOL Usage   PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
  GEN          MMMMMMMMMMMMMMMMMMMMM
  CKPT Space
  DB0: Busy    DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
  DB0: Blks    BBBBBBBBBBBBBBB

               0         +       20          +       40          +        60         +        80         +       100  (% utilization)

Data Collection Ended   15-JUN-81 09:50:02.9 002703 035435
   SECONDS (System-wide)    Elapsed=965.41303   Kernel=179.76893    Idle=360.49064   Device=280.11297


Data Reduction Period (Seconds) = 965.41303
```

```
RSX-11M Performance Monitor Data Reduction V01.000        18-JUN-81 14:51:07         Page 5        Task Summary

                          NEW VERSION 8-JUN-81

Generic TASK = QT0000

Elapsed Times (seconds):       244.0             244.0             244.0            0.00             243.956
                             minimum             mean            maximum          coef(var)            total

Number of Task Invocations = 1                ( 1 complete )              ( 0 incomplete )

Memory Times     seconds:    215.6        28.39        28.39        % elapsed:     88.4          11.6          11.6
                               in          out          wait                         in           out          wait

Memory Size in KWords:        17.47        17.47        17.47
                               min          mean          max

Space-Time Product:            13      3765.68        289.668        1.60
                         increments    total KWS      mean KWS     coef(var) KWS    incomplete

ACP Counts:           0              0                0               0               0                0              0
      access/deaccess     create/delete       read/write       directory   file attribute          extend          other

    RESOURCE                 +- - - - - - - - - - - - - - - - - - USAGE  TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                       count   total time    % resident time     minimum        mean        maximum       coef(var)       % resource
Memory Residency         13     215.567           N/A             7.9617       16.5821      108.5648        1.60            N/A
CPU Timeslices          744     50.9689          23.6             0.0002        0.0685        0.2671        0.88            5.3
Checkpoint R + W         24     1.62545           0.8             0.0565        0.0677        0.1022        0.15            0.2
Initial Loads             1    0.687000E-01       0.0             0.0687        0.0687        0.0687        0.00            0.0

    RESOURCE                 +- - - - - - - - - - - - - - - - - - WAIT   TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                       count   total time    % resident time     minimum        mean        maximum       coef(var)
Memory Residency         13    28.3886            N/A             0.0003        2.1837        4.5347        0.74
Checkpoint R + W         24    0.367550           0.2             0.0004        0.0153        0.0869        1.74
Initial Loads             1    0.102000E-02       0.0             0.0010        0.0010        0.0010        0.00

    RESOURCE                 +- - - - - - - - - - - - - - - - - SERVICE  TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                       count   total time    % resident time     minimum        mean        maximum       coef(var)
Memory Residency         13     243.956           N/A             7.9620       18.7658      110.0780        1.41
Checkpoint R + W         24     1.99300           0.9             0.0574        0.0830        0.1558        0.38
Initial Loads             1    0.697200E-01       0.0             0.0697        0.0697        0.0697        0.00

    RESOURCE                      +- - - - - -  ARRIVALS  PER  SECOND  - - - - - -+
                     # arrivals          mean task rate          mean system rate
Memory Residency         13                   0.05                      0.01
CPU Timeslices          744                   3.05                      0.77
Checkpoint R + W         24                   0.10                      0.02
Initial Loads             1                   0.00                      0.00
```

```
RSX-11M Performance Monitor Data Reduction V01.000          18-JUN-81 14:51:07          Page 6          Task Summary

                               NEW VERSION 8-JUN-81

Generic TASK = QT0100

Elapsed Times (seconds):       279.1              279.1              279.1              0.00              279.138
                              minimum             mean              maximum           coef(var)            total

Number of Task Invocations = 1                  ( 1 complete )                  ( 0 incomplete )

Memory Times      seconds:    173.7         105.4        105.4        % elapsed:     62.2          37.8          37.8
                                in           out          wait                        in           out           wait

Memory Size in KWords:        17.47        17.47        17.47
                               min          mean          max

Space-Time Product:            15      3034.66        202.311        0.88
                          increments   total KWS      mean KWS    coef(var) KWS    incomplete

ACP Counts:            0               0                0               0               0                0               0
       access/deaccess     create/delete       read/write       directory    file attribute       extend          other

    RESOURCE                   +- - - - - - - - - - - - - - - - - - USAGE  TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                      count     total time    % resident time    minimum        mean         maximum        coef(var)      % resource
Memory Residency         15      173.719            N/A           7.8162       11.5813       49.8459          0.88            N/A
CPU Timeslices          541      48.5559           28.0           0.0002        0.0898        0.2675          0.85            5.0
Checkpoint R + W         28      1.84705            1.1           0.0565        0.0660        0.0852          0.10            0.2
Initial Loads             1     0.609800E-01        0.0           0.0610        0.0610        0.0610          0.00            0.0

    RESOURCE                   +- - - - - - - - - - - - - - - - - - WAIT   TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                      count     total time    % resident time    minimum        mean         maximum        coef(var)
Memory Residency         15      105.419            N/A           0.0003        7.0279       71.4317          2.46
Checkpoint R + W         28     0.388570            0.2           0.0009        0.0139        0.0906          1.80
Initial Loads             1     0.102000E-02        0.0           0.0010        0.0010        0.0010          0.00

    RESOURCE                   +- - - - - - - - - - - - - - - - - SERVICE  TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                      count     total time    % resident time    minimum        mean         maximum        coef(var)
Memory Residency         15      279.138            N/A           7.8165       18.6092      121.2776          1.48
Checkpoint R + W         28      2.23562            1.3           0.0574        0.0798        0.1482          0.34
Initial Loads             1     0.620000E-01        0.0           0.0620        0.0620        0.0620          0.00

    RESOURCE                          +- - - - - -  ARRIVALS  PER  SECOND  - - - - - -+
                    # arrivals            mean task rate            mean system rate
Memory Residency         15                    0.05                      0.02
CPU Timeslices          541                    1.94                      0.56
Checkpoint R + W         28                    0.10                      0.03
Initial Loads             1                    0.00                      0.00
```

```
RSX-11M Performance Monitor Data Reduction V01.000        18-JUN-81 14:51:07         Page 7         Task Summary

                            NEW VERSION 8-JUN-81

Generic TASK = QT0200

Elapsed Times (seconds):       268.7              268.7              268.7              0.00              268.726
                              minimum              mean             maximum          coef(var)             total

Number of Task Invocations = 1                    ( 1 complete )                   ( 0 incomplete )

Memory Times      seconds:     228.7         40.06          40.06         % elapsed:      85.1          14.9          14.9
                                in            out           wait                          in            out           wait

Memory Size in KWords:         17.47         17.47          17.47
                                min           mean           max

Space-Time Product:             15      3994.58         266.305           1.68
                          increments    total KWS        mean KWS       coef(var) KWS    incomplete

ACP Counts:              0             0              0              0              0               0              0
       access/deaccess     create/delete      read/write      directory     file attribute        extend          other

  RESOURCE                 +- - - - - - - - - - - - - - - - - - USAGE  TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                        count    total time     % resident time     minimum       mean        maximum       coef(var)     % resource
Memory Residency          15     228.670             N/A            0.6066       15.2447      110.9225        1.68            N/A
CPU Timeslices           635      48.7575           21.3            0.0002        0.0768        0.2680        0.70            5.1
Checkpoint R + W          28       1.89335           0.8            0.0566        0.0676        0.0972        0.14            0.2
Initial Loads              1     0.737900E-01        0.0            0.0738        0.0738        0.0738        0.00            0.0

  RESOURCE                 +- - - - - - - - - - - - - - - - - - WAIT   TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                        count    total time     % resident time     minimum       mean        maximum       coef(var)
Memory Residency          15      40.0557            N/A            0.1427        2.6704        6.0437        0.68
Checkpoint R + W          28     0.429260            0.2            0.0007        0.0153        0.0981        1.81
Initial Loads              1     0.119000E-02        0.0            0.0012        0.0012        0.0012        0.00

  RESOURCE                 +- - - - - - - - - - - - - - - - - SERVICE  TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                        count    total time     % resident time     minimum       mean        maximum       coef(var)
Memory Residency          15     268.726             N/A            1.9748       17.9151      116.9663        1.49
Checkpoint R + W          28       2.32261           1.0            0.0575        0.0830        0.1643        0.36
Initial Loads              1     0.749800E-01        0.0            0.0750        0.0750        0.0750        0.00

  RESOURCE                        +- - - - - -  ARRIVALS  PER  SECOND  - - - - - -+
                      # arrivals          mean task rate          mean system rate
Memory Residency          15                   0.06                     0.02
CPU Timeslices           635                   2.36                     0.66
Checkpoint R + W          28                   0.10                     0.03
Initial Loads              1                   0.00                     0.00
```

```
RSX-11M Performance Monitor Data Reduction V01.000        18-JUN-81 14:51:07        Page 8        Task Summary

                              NEW VERSION 8-JUN-81

Generic TASK = QT0300

Elapsed Times (seconds):      229.2              229.2              229.2              0.00              229.200
                             minimum             mean              maximum           coef(var)           total

Number of Task Invocations = 1                ( 1 complete )                     ( 0 incomplete )

Memory Times     seconds:     203.6          25.57          25.57          % elapsed:      88.8          11.2          11.2
                              in             out            wait                           in            out           wait

Memory Size in KWords:        17.47          17.47          17.47
                              min            mean           max

Space-Time Product:               12      3557.21         296.434          1.55
                          increments      total KWS       mean KWS        coef(var) KWS     incomplete

ACP Counts:             0               0                0                  0                0                 0               0
          access/deaccess      create/delete        read/write         directory    file attribute          extend           other

     RESOURCE                  +- - - - - - - - - - - - - - - - - - - - USAGE  TIMES  IN  SECONDS  - - - - - - - - - - - - - - -+
                        count     total time     % resident time     minimum       mean        maximum       coef(var)      % resource
Memory Residency           12      203.633           N/A              7.7787      16.9694     104.3405        1.55              N/A
CPU Timeslices            621      48.7566           23.9             0.0002       0.0785       0.2680        0.84              5.1
Checkpoint R + W           22      1.47733            0.7             0.0566       0.0672       0.1004        0.14              0.2
Initial Loads               1     0.789900E-01        0.0             0.0790       0.0790       0.0790        0.00              0.0

     RESOURCE                  +- - - - - - - - - - - - - - - - - - - - WAIT   TIMES  IN  SECONDS  - - - - - - - - - - - - - - -+
                        count     total time     % resident time     minimum       mean        maximum       coef(var)
Memory Residency           12      25.5669           N/A              0.1415       2.1306       4.4752        0.73
Checkpoint R + W           22     0.335170            0.2             0.0005       0.0152       0.1018        1.89
Initial Loads               1     0.137000E-02        0.0             0.0014       0.0014       0.0014        0.00

     RESOURCE                  +- - - - - - - - - - - - - - - - - - - SERVICE  TIMES  IN  SECONDS  - - - - - - - - - - - - - - -+
                        count     total time     % resident time     minimum       mean        maximum       coef(var)
Memory Residency           12      229.200           N/A              8.9906      19.1000     104.4828        1.35
Checkpoint R + W           22      1.81250            0.9             0.0575       0.0824       0.1634        0.37
Initial Loads               1     0.803600E-01        0.0             0.0804       0.0804       0.0804        0.00

     RESOURCE                         +- - - - - -  ARRIVALS  PER  SECOND  - - - - - -+
                     # arrivals             mean task rate            mean system rate
Memory Residency           12                    0.05                        0.01
CPU Timeslices            621                    2.71                        0.64
Checkpoint R + W           22                    0.10                        0.02
Initial Loads               1                    0.00                        0.00
```

```
RSX-11M Performance Monitor Data Reduction V01.000         18-JUN-81 14:51:07           Page 9         Task Summary

                                  NEW VERSION 8-JUN-81

Generic TASK = QT0400

Elapsed Times (seconds):      268.9               268.9               268.9               0.00               268.854
                            minimum               mean              maximum            coef(var)             total

Number of Task Invocations = 1                 ( 1 complete )                  ( 0 incomplete )

Memory Times      seconds:    233.4         35.42          35.42        % elapsed:     86.8          13.2           13.2
                                in           out            wait                        in           out            wait

Memory Size in KWords:         17.47        17.47        17.47
                                min          mean          max

Space-Time Product:            14      4077.75          291.268           1.67
                       increments    total KWS          mean KWS       coef(var) KWS    incomplete

ACP Counts:           0                  0                  0                  0                  0                  0                  0
       access/deaccess     create/delete         read/write          directory    file attribute           extend              other

    RESOURCE                    +- - - - - - - - - - - - - - - - - USAGE  TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                      count     total time   % resident time    minimum          mean        maximum         coef(var)      % resource
Memory Residency         14      233.431            N/A          8.4057        16.6737      117.1903          1.67            N/A
CPU Timeslices          675      50.0753           21.5          0.0002         0.0742        0.2672           0.73            5.2
Checkpoint R + W         26      1.72913            0.7          0.0565         0.0665        0.0799           0.09            0.2
Initial Loads             1     0.738500E-01        0.0          0.0739         0.0739        0.0739           0.00            0.0

    RESOURCE                    +- - - - - - - - - - - - - - - - - WAIT   TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                      count     total time   % resident time    minimum          mean        maximum         coef(var)
Memory Residency         14      35.4224            N/A          0.1377         2.5302        7.6624           0.84
Checkpoint R + W         26     0.396470            0.2          0.0005         0.0152        0.1054           1.80
Initial Loads             1     0.713400E-01        0.0          0.0713         0.0713        0.0713           0.00

    RESOURCE                    +- - - - - - - - - - - - - - - - SERVICE  TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                      count     total time   % resident time    minimum          mean        maximum         coef(var)
Memory Residency         14      268.854            N/A          8.9074        19.2038      120.1552          1.46
Checkpoint R + W         26      2.12560            0.9          0.0572         0.0818        0.1746           0.36
Initial Loads             1     0.145190            0.1          0.1452         0.1452        0.1452           0.00

    RESOURCE                          +- - - - - -  ARRIVALS  PER  SECOND  - - - - - -+
                     # arrivals           mean task rate            mean system rate
Memory Residency         14                   0.05                       0.01
CPU Timeslices          675                   2.51                       0.70
Checkpoint R + W         26                   0.10                       0.03
Initial Loads             1                   0.00                       0.00
```

```
RSX-11M Performance Monitor Data Reduction V01.000         18-JUN-81 14:51:07          Page 10         Task Summary
                                  NEW VERSION 8-JUN-81

Generic TASK = SHF...

Elapsed Times (seconds):      0.2200E-02          0.8924E-01           1.305               1.12               50.9579
                             minimum              mean               maximum            coef(var)            total

Number of Task Invocations = 571                    ( 571 complete )                   ( 0 incomplete )

ACP Counts:          0                0                0                0                0                0                0
       access/deaccess     create/delete        read/write        directory     file attribute          extend           other

    RESOURCE                   +- - - - - - - - - - - - - - - - - - USAGE  TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                       count    total time    % resident time     minimum         mean         maximum       coef(var)      % resource
CPU Timeslices           1263    4.86158                           0.0002         0.0038        0.2998         4.35              0.5

    RESOURCE                   +- - - - - - - - - - - - - - - - - - WAIT   TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                       count    total time    % resident time     minimum         mean         maximum       coef(var)

    RESOURCE                   +- - - - - - - - - - - - - - - - - SERVICE  TIMES  IN  SECONDS  - - - - - - - - - - - - - -+
                       count    total time    % resident time     minimum         mean         maximum       coef(var)

    RESOURCE                         +- - - - - -  ARRIVALS  PER  SECOND  - - - - - -+
                     # arrivals           mean task rate           mean system rate
CPU Timeslices           1263                  24.79                        1.31
```

```
RSX-11M Performance Monitor Data Reduction V01.000        18-JUN-81 14:51:07        Page 11        Task Summary

                                  NEW VERSION 8-JUN-81

Generic TASK = ...LDR

Elapsed Times (seconds):      960.2              960.2              960.2              0.00              960.209
                             minimum             mean              maximum           coef(var)            total

Number of Task Invocations = 1                 ( 0 complete )                  ( 1 incomplete )

ACP Counts:          0             0              0              0              0              0              0
      access/deaccess     create/delete      read/write      directory    file attribute      extend         other

    RESOURCE              +- - - - - - - - - - - - - - - - - - USAGE  TIMES  IN  SECONDS - - - - - - - - - - - - - -+
                    count    total time    % resident time     minimum       mean        maximum       coef(var)      % resource
CPU Timeslices       1629     3.51720                           0.0013      0.0022       0.0133         0.42              0.4
DB0: QIOs             914    32.4975                            0.0081      0.0356       0.1080         0.43              3.4

    RESOURCE              +- - - - - - - - - - - - - - - - - - WAIT   TIMES  IN  SECONDS - - - - - - - - - - - - - -+
                    count    total time    % resident time     minimum       mean        maximum       coef(var)
DB0: QIOs             914    0.917460                           0.0003      0.0010       0.0316         3.55

    RESOURCE              +- - - - - - - - - - - - - - - - - SERVICE TIMES  IN  SECONDS - - - - - - - - - - - - - -+
                    count    total time    % resident time     minimum       mean        maximum       coef(var)
DB0: QIOs             914    33.4150                            0.0085      0.0366       0.1083         0.44

    RESOURCE                    +- - - - - -  ARRIVALS  PER  SECOND  - - - - - -+
                  # arrivals           mean task rate          mean system rate
CPU Timeslices       1629                   1.70                      1.69
DB0: QIOs             914                   0.95                      0.95
```

```
RSX-11M Performance Monitor Data Reduction V01.000          18-JUN-81 14:51:07          Page 12          Task Summary

                                       NEW VERSION 8-JUN-81

Generic TASK = ...SIO

Elapsed Times (seconds):       191.6               191.6               191.6               0.00               191.611
                             minimum               mean              maximum            coef(var)              total

Number of Task Invocations = 1                    ( 1 complete )                      ( 0 incomplete )

Memory Times      seconds:    191.6        0.5100E-03      0.2800E-03     % elapsed:    100.0          0.0           0.0
                              in           out             wait                         in             out           wait

Memory Size in KWords:         14.00        14.00           14.00
                               min          mean            max

Space-Time Product:                   1      2682.55         2682.55          0.00
                             increments    total KWS        mean KWS      coef(var) KWS    incomplete

ACP Counts:             12             10            1133              2              0               11               0
       access/deaccess       create/delete       read/write        directory   file attribute       extend           other

    RESOURCE                    +- - - - - - - - - - - - - - - - - - USAGE  TIMES  IN  SECONDS - - - - - - - - - - - - - -+
                        count    total time    % resident time     minimum        mean         maximum      coef(var)      % resource
Memory Residency            1     191.611            N/A           191.6108      191.6108      191.6108        0.00            N/A
CPU Timeslices           3414     129.090           67.4             0.0002        0.0378        0.2293        0.98           13.4
Initial Loads               1    0.729700E-01        0.0             0.0730        0.0730        0.0730        0.00            0.0
DBO: Ovly Loads           241     2.80417            1.5             0.0019        0.0116        0.0338        0.45            0.3
DBO: QIOs                6989     117.883           61.5             0.0019        0.0169        0.0360        0.35           12.2
DB1: QIOs                3778     43.6882           22.8             0.0014        0.0116        0.0351        0.46            4.5
DSO: QIOs                7426     75.4969           39.4             0.0014        0.0102        0.0251        0.49            7.8
DS1: QIOs                7426     74.0504           38.6             0.0014        0.0100        0.0249        0.50            7.7
MMO: QIOs                   2     186.205           97.2             0.0005       93.1023      186.2040        1.00           19.3
F11ACP QIOs              1149     6.50660            3.4             0.0011        0.0057        0.3712        3.48            0.7
DS1***ACP QIOs             11    0.592160            0.3             0.0047        0.0538        0.1113        0.57            0.1
DSO***ACP QIOs              8    0.392530            0.2             0.0044        0.0491        0.1290        0.77            0.0

    RESOURCE                    +- - - - - - - - - - - - - - - - - - WAIT   TIMES  IN  SECONDS - - - - - - - - - - - - - -+
                        count    total time    % resident time     minimum        mean         maximum      coef(var)
Memory Residency            1    0.280000E-03        N/A             0.0003        0.0003        0.0003        0.00
Initial Loads               1    0.115000E-02        0.0             0.0011        0.0011        0.0011        0.00
CBO: Ovly Loads           241    0.190430            0.1             0.0003        0.0008        0.0342        4.42
DBO: QIOs                6989     176.900           92.3             0.0003        0.0253        0.1329        0.59
DB1: QIOs                3778     93.1276           48.6             0.0003        0.0246        0.0986        0.63
DSO: QIOs                7426     41.9457           21.9             0.0003        0.0056        0.0226        0.86
DS1: QIOs                7426     43.2159           22.6             0.0003        0.0058        0.0225        0.84
MMO: QIOs                   2    0.610000E-03        0.0             0.0003        0.0003        0.0003        0.02
F11ACP QIOs              1149     1.39404            0.7             0.0010        0.0012        0.0086        0.53
DS1***ACP QIOs             11    0.396300E-01        0.0             0.0010        0.0036        0.0282        2.16
DSO***ACP QIOs              8    0.912000E-02        0.0             0.0010        0.0011        0.0020        0.31

    RESOURCE                    +- - - - - - - - - - - - - - - - - SERVICE  TIMES  IN  SECONDS - - - - - - - - - - - - - -+
                        count    total time    % resident time     minimum        mean         maximum      coef(var)
Memory Residency            1     191.611            N/A           191.6111      191.6111      191.6111        0.00
Initial Loads               1    0.741200E-01        0.0             0.0741        0.0741        0.0741        0.00
DBO: Ovly Loads           241     2.99460            1.6             0.0022        0.0124        0.0538        0.55
```

```
RSX-11M Performance Monitor Data Reduction V01.000         18-JUN-81 14:51:07         Page 13        Task Summary

                                     NEW VERSION 8-JUN-81

DB0: QIOs               6989    294.784           153.8            0.0022        0.0422        0.1488          0.41
DB1: QIOs               3778    136.816            71.4            0.0022        0.0362        0.1103          0.43
DS0: QIOs               7426    117.443            61.3            0.0017        0.0158        0.0344          0.47
DS1: QIOs               7426    117.266            61.2            0.0018        0.0158        0.0367          0.47
MM0: QIOs                  2    186.205            97.2            0.0008       93.1026      186.2044          1.00
F11ACP QIOs             1149    7.90064             4.1            0.0021        0.0069        0.3721          2.86
DS1***ACP QIOs            11   0.631790             0.3            0.0057        0.0574        0.1126          0.54
DS0***ACP QIOs             8   0.401650             0.2            0.0053        0.0502        0.1303          0.76


    RESOURCE                            +- - - - - -  ARRIVALS  PER  SECOND  - - - - - -+
                       # arrivals             mean task rate            mean system rate
Memory Residency            1                      0.01                      0.00
CPU Timeslices           3414                     17.82                      3.54
Initial Loads               1                      0.01                      0.00
DB0: Ovly Loads           241                      1.26                      0.25
DB0: QIOs                6989                     36.47                      7.24
DB1: QIOs                3778                     19.72                      3.91
DS0: QIOs                7426                     38.76                      7.69
DS1: QIOs                7426                     38.76                      7.69
MM0: QIOs                   2                      0.01                      0.00
F11ACP QIOs              1149                      6.00                      1.19
DS1***ACP QIOs             11                      0.06                      0.01
DS0***ACP QIOs              8                      0.04                      0.01
```

```
RSX-11M Performance Monitor Data Reduction V01.000        18-JUN-81 14:51:07        Page 14        System-Wide Summary

                               NEW VERSION 8-JUN-81

Elapsed Times (seconds):    0.2200E-02         4.313           960.2          10.92           2492.65
                            minimum            mean            maximum        coef(var)       total

Number of Task Invocations = 578              ( 577 complete )                ( 1 incomplete )

Memory Times     seconds:   1247.       1246.        234.9        % elapsed:    50.0       50.0       9.4
                            in          out          wait                       in         out        wait

Memory Size in KWords:      14.00       16.94        17.47
                            min         mean         max

Space-Time Product:             70      21112.4        301.606        1.68
                          increments    total KWS      mean KWS       coef(var) KWS    incomplete

ACP Counts:          12               10             1133              2              0             11            0
        access/deaccess      create/delete      read/write       directory    file attribute     extend        other

   RESOURCE                     +- - - - - - - - - - - - - - - - - - USAGE  TIMES  IN  SECONDS - - - - - - - - - - - - - -+
                        count     total time    % resident time    minimum      mean       maximum      coef(var)     % resource
Memory Residency          70      1246.63           N/A            0.6066     17.8090     191.6108       1.79           N/A
CPU Timeslices          9522      384.583          30.8            0.0002      0.0404       0.2998       1.29          39.8
Checkpoint R + W         128      8.57231           0.7            0.0565      0.0670       0.1022       0.13           0.9
Initial Loads              6     0.429280           0.0            0.0610      0.0715       0.0790       0.08           0.0
DB0: Ovly Loads          241      2.80417           0.2            0.0019      0.0116       0.0338       0.45           0.3
DB0: QIOs               7903      150.381          12.1            0.0019      0.0190       0.1080       0.51          15.6
DB1: QIOs               3778      43.6882           3.5            0.0014      0.0116       0.0351       0.46           4.5
DS0: QIOs               7426      75.4969           6.1            0.0014      0.0102       0.0251       0.49           7.8
DS1: QIOs               7426      74.0504           5.9            0.0014      0.0100       0.0249       0.50           7.7
MM0: QIOs                  2      186.205          14.9            0.0005     93.1023     186.2040       1.00          19.3
F11ACP QIOs             1149      6.50660           0.5            0.0011      0.0057       0.3712       3.48           0.7
DS1***ACP QIOs            11     0.592160           0.0            0.0047      0.0538       0.1113       0.57           0.1
DS0***ACP QIOs             8     0.392530           0.0            0.0044      0.0491       0.1290       0.77           0.0

   RESOURCE                     +- - - - - - - - - - - - - - - - - - WAIT   TIMES  IN  SECONDS - - - - - - - - - - - - - -+
                        count     total time    % resident time    minimum      mean       maximum      coef(var)
Memory Residency          70      234.853           N/A            0.0003      3.3550      71.4317       2.50
Checkpoint R + W         128      1.91702           0.2            0.0004      0.0150       0.1054       1.81
Initial Loads              6    0.770900E-01        0.0            0.0010      0.0128       0.0713       2.04
3B0: Ovly Loads          241     0.190430           0.0            0.0003      0.0008       0.0342       4.42
DB0: QIOs               7903      177.818          14.3            0.0003      0.0225       0.1329       0.72
DB1: QIOs               3778      93.1276           7.5            0.0003      0.0246       0.0986       0.63
DS0: QIOs               7426      41.9457           3.4            0.0003      0.0056       0.0226       0.86
DS1: QIOs               7426      43.2159           3.5            0.0003      0.0058       0.0225       0.84
MM0: QIOs                  2    0.610000E-03        0.0            0.0003      0.0003       0.0003       0.02
F11ACP QIOs             1149      1.39404           0.1            0.0010      0.0012       0.0086       0.53
DS1***ACP QIOs            11    0.396300E-01        0.0            0.0010      0.0036       0.0282       2.16
DS0***ACP QIOs             8    0.912000E-02        0.0            0.0010      0.0011       0.0020       0.31

   RESOURCE                     +- - - - - - - - - - - - - - - - - SERVICE  TIMES  IN  SECONDS - - - - - - - - - - - - - -+
                        count     total time    % resident time    minimum      mean       maximum      coef(var)
Memory Residency          70      1481.48           N/A            1.9748     21.1641     191.6111       1.59
Checkpoint R + W         128      10.4893           0.8            0.0572      0.0819       0.1746       0.36
Initial Loads              6     0.506370           0.0            0.0620      0.0844       0.1452       0.33
```

```
RSX-11M Performance Monitor Data Reduction V01.000        18-JUN-81 14:51:07        Page 15        System-Wide Summary

                                  NEW VERSION 8-JUN-81

DB0: Ovly Loads        241    2.99460          0.2          0.0022      0.0124      0.0538      0.55
DB0: QIOs             7903    328.199         26.3          0.0022      0.0415      0.1488      0.41
DB1: QIOs             3778    136.816         11.0          0.0022      0.0362      0.1103      0.43
DS0: QIOs             7426    117.443          9.4          0.0017      0.0158      0.0344      0.47
DS1: QIOs             7426    117.266          9.4          0.0018      0.0158      0.0367      0.47
MM0: QIOs                2    186.205         14.9          0.0008     93.1026    186.2044      1.00
F11ACP QIOs           1149    7.90064          0.6          0.0021      0.0069      0.3721      2.86
DS1***ACP QIOs          11   0.631790          0.1          0.0057      0.0574      0.1126      0.54
DS0***ACP QIOs           8   0.401650          0.0          0.0053      0.0502      0.1303      0.76

    RESOURCE                            +- - - - - -  ARRIVALS  PER  SECOND  - - - - - -+
                      # arrivals             mean task rate          mean system rate
Memory Residency          70                    0.03                      0.07
CPU Timeslices          9522                    3.82                      9.86
Checkpoint R + W         128                    0.05                      0.13
Initial Loads              6                    0.00                      0.01
DB0: Ovly Loads          241                    0.10                      0.25
DB0: QIOs               7903                    3.17                      8.19
DB1: QIOs               3778                    1.52                      3.91
DS0: QIOs               7426                    2.98                      7.69
DS1: QIOs               7426                    2.98                      7.69
MM0: QIOs                  2                    0.00                      0.00
F11ACP QIOs             1149                    0.46                      1.19
DS1***ACP QIOs            11                    0.00                      0.01
DS0***ACP QIOs             8                    0.00                      0.01
```

```
RSX-11M Performance Monitor Data Reduction V01.000        18-JUN-81 14:51:07        Page 16        Reduction Run Statistics
                                           NEW VERSION 8-JUN-81


528. Buffers Read
134273. Records Read
0. Buffers with Overflow
0. Records Lost by Overflow
```

Appendix G
EXAMPLE SMPGEN

```
>SET /UIC=[11,2]
>@SMPGEN
>;
>;*** Software Monitor Parameter GENeration command file ***
>;
>;*** Version 1 ***
>;
>;Copyright (C) 1981 Digital Equipment Corporation, Maynard, Mass.
>;
>;
>; Initialize default parameters
>;
>* READ SAVED ANSWER FILE FOR DEFAULT ANSWERS? [Y/N]:N
>;
>; Select parameter category, type <esc> for help
>;
>* ENTER PARAMETER CATEGORY [S]:
>;
>; LEGAL PARAMETER CATEGORIES ARE :
>;       HOOK             TASK             MODE             LABEL
>;       BUFFER           DEVICE           SYSTEM           DONE
>;
>* ENTER PARAMETER CATEGORY [S]: HOOK
>;
>; Select/deselect hook points
>;
>* CPU USAGE HOOKS? [Y/N]:Y
>* MEMORY USAGE HOOKS? [Y/N]:N
>* I/O DEVICE USAGE HOOKS? [Y/N]:Y
>;
>; Select parameter category, type <esc> for help
>;
>* ENTER PARAMETER CATEGORY [S]: BUFFER
>;
>; Setup buffer management parameter, type <esc> for help
>;
>* ENTER BUFFER TYPE [D:DYNAMIC] [S R:0-7]:
>* ENTER NAME OF BUFFER MAIN (SYS) PARTITION [D:GEN] [S R:0-6]:
>* ENTER NAME OF BUFFER COMMON/REGION [D:TRCPAR] [S R:0-6]:
>* ENTER NUMBER OF BUFFERS [D R:2.-8. D:2.]: 3
>* ENTER BUFFER SIZE IN 512B BLOCKS [D R:1.-16. D:8.]: 6
>;
>; Select parameter category, type <esc> for help
>;
>* ENTER PARAMETER CATEGORY [S]: TASK
>;
>; Select/deselect tasks to be traced
>;
>* DO YOU WANT TO TRACE ALL TASKS? [Y/N]:N
>* DO YOU WANT TO SPECIFY THE TASKS NOT TO TRACE? [Y/N]:Y
>;
>; Enter one taskname per line. Terminate list by <CR>.
>;
>* ENTER NAME OF TASK NOT TO TRACE [S]: F11ACP
>* ENTER NAME OF TASK NOT TO TRACE [S]: MCR...
>* ENTER NAME OF TASK NOT TO TRACE [S]: QMG...
>* ENTER NAME OF TASK NOT TO TRACE [S]: LPP0
>* ENTER NAME OF TASK NOT TO TRACE [S]:
>;
>; Select parameter category, type <esc> for help
>;
>* ENTER PARAMETER CATEGORY [S]: DEVICE
>* DO YOU WANT TO COLLECT DATA FOR ALL DEVICES? [Y/N]:N
>;
>; Select device types to be traced
>;
>* TRACE DISK (DIRECTORY DEVICE) I/O ACTIVITY? [Y/N]:Y
>* TRACE TAPE (SEQUENTIAL DEVICE) I/O ACTIVITY? [Y/N]:Y
>* TRACE TERMINAL I/O ACTIVITY? [Y/N]:N
>* TRACE ALL OTHER DEVICES I/O ACTIVITY? [Y/N]:N
>;
>; Select parameter category, type <esc> for help
>;
>* ENTER PARAMETER CATEGORY [S]: MODE
>;
>; Define data collection mode, type <esc> for help
>;
>* ENTER MODE [D:MANUAL] [S R:0.-8.]:
>;
>;
>; LEGAL MODES ARE:
>;      MANUAL            AUTOSTOP          REPEAT
>;
>* ENTER MODE [D:MANUAL] [S R:0.-8.]: AUTOSTOP
>* ENTER SAMPLE TYPE [S R:0-6]:
>;
>; LEGAL SAMPLE TYPES ARE:
>;
>;      BUFFER            TIME
>;
>* ENTER SAMPLE TYPE [S R:0-6]: TIME
>* ENTER SAMPLING INTERVAL TIME UNIT [D:MIN] [S R:0-3]: HR
>* ENTER SAMPLING INTERVAL IN HRS [D R:1.-10000. D:1.]: 2
>;
>; Select parameter category, type <esc> for help
>;
>* ENTER PARAMETER CATEGORY [S]: SYSTEM
>;
>; Select system metrics
>;
>* DO YOU WANT THE SYSTEM-WIDE METRICS? [Y/N]:Y
>* ENTER SYSTEM METRICS SAMPLING INTERVAL (SECS) [D R:15.-28800. D:60.]: 120.
>;
>; Select parameter category, type <esc> for help
>;
>* ENTER PARAMETER CATEGORY [S]: LABEL
>;
>; Define 1 to 63 character label string, type <esc> for help
>;
>* ENTER MPB LABEL [S R:0-63]: THIS IS A SAMPLE MPB GENERATION
>;
>; Select parameter category, type <esc> for help
>;
>* ENTER PARAMETER CATEGORY [S]: DONE
>;
>; Write saved answer file and parameter source file
>; Enter <esc> if no output file is desired
>;
>* ENTER NAME OF SAVED ANSWER FILE [D:TRACE.CMD] [S]: SAMPLE.CMD
>* ENTER NAME OF PARAMETER SOURCE FILE [D:TRACE.PAR] [S]: SAMPLE.PAR
>MAC SAMPLE=[11,10]RSXMC,[11,2]HKMAC,[11,2]SAMPLE.PAR
>TKB @PARBLD
>PIP SAMPLE.OBJ;*/DE,PARBLD.CMD;*
>@ <EOF>
>
```