

PERKIN-ELMER

**OS/32
MULTI-TERMINAL MONITOR (MTM)
SYSTEM PLANNING AND OPERATOR**

Reference Manual

48-023 F00 R03

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

The software described in this document is furnished under a license, and it can be used or copied only in a manner permitted by that license. Any copy of the described software must include the Perkin-Elmer copyright notice. Title to and ownership of the described software and any copies thereof shall remain in The Perkin-Elmer Corporation.

The Perkin-Elmer Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Perkin-Elmer.

The Perkin-Elmer Corporation, Data Systems Group, 2 Crescent Place, Oceanport, New Jersey 07757

© 1977, 1981, 1983, 1984 by The Perkin-Elmer Corporation

Printed in the United States of America

TABLE OF CONTENTS

PREFACE		vii	
CHAPTERS			
1	GENERAL DESCRIPTION		
1.1	INTRODUCTION	1-1	
1.2	MULTI-TERMINAL MONITOR (MTM) SYSTEM REQUIREMENTS	1-2	
1.3	CONFIGURING A MULTI-TERMINAL MONITOR (MTM) SYSTEM	1-2	
1.4	OS/32 SYSTEM GENERATION (SYSGEN) OPTIONS	1-3	
1.5	FILE DESCRIPTORS	1-3	
2	MULTI-TERMINAL MONITOR (MTM) SYSTEM GENERATION (SYSGEN)		
2.1	INTRODUCTION	2-1	
2.2	SELECTING MULTI-TERMINAL MONITOR (MTM) SYSTEM GENERATION (SYSGEN) OPTIONS	2-1	
2.3	GENERATING A MULTI-TERMINAL MONITOR (MTM) SYSTEM USING THE STANDARD COMMAND SUBSTITUTION SYSTEM (CSS) COMMAND FILE (MTMSGN.CSS)	2-4	
2.4	MULTI-TERMINAL MONITOR (MTM) SYSTEM GENERATION (SYSGEN) OPTIONS	2-5	
2.4.1	Sector Buffer Size (CDBUFSIZ) Option	2-6	
2.4.2	Special Console Buffer Size (CONBFSZ) Option	2-7	
2.4.3	Command Buffer Size (CSBFSZ) Option	2-7	
2.4.4	Input/Output (I/O) Class Table (IOTAB) Macro	2-8	
2.4.5	Multi-Terminal Monitor (MTM) Link (LINKOPT) Option	2-9	
2.4.6	Number of Special Console Buffers (NCONBFS) Option	2-10	
2.4.7	Number of Command Buffers (NCSBFS) Option	2-12	
2.4.8	Number of Command Substitution System (CSS) Levels (NCSSLVS) Option	2-14	

CHAPTERS (Continued)

2.4.9	Number of Message Buffers (NMSGs) Option	2-14
2.4.10	Number of Sector Length Buffers (NSECTS) Option	2-15
2.4.11	Number of Concurrent Users (NUSERS) Option	2-16
2.4.12	Batch Job Processor Time Limit (PRCLIMIT) Option	2-16
2.4.13	Multi-Terminal Monitor (MTM) Terminal Prompt Device Name (PRMTDEV) Macro	2-17
2.4.14	Multi-Terminal Monitor (MTM) Terminal Prompt Sequence (PRMTSEQ) Macro	2-17
2.4.15	Multi-Terminal Monitor (MTM) Task Queue Size (QUESIZE) Option	2-19
2.4.16	OS/32 Accounting Facility (SGN.ACT) Option	2-19
2.4.17	Batch Job Pause (SGN.BTCP) Option	2-20
2.4.18	Default User Volume (SGN.MVOL) Option	2-20
2.4.19	Batch Job Relative Priority (SGN.PRIO) Option	2-21
2.4.20	Forced Roll (SGN.ROLL) Option	2-23
2.4.21	Multi-Terminal Monitor (MTM) Scheduling (SGN.SCHR) Option	2-23
2.4.22	SPL/32 Interface (SGN.SPL) Option	2-27
2.4.23	Global and Local Variables (SGN.VAR) Option	2-27
2.4.24	Number of New Global Variables (SGN.@GLO) Option	2-28
2.4.25	Number of New Internal Variables (SGN.@INT) Option	2-28
2.4.26	HASP Interface (SGN.\$HSP) Option	2-29
2.4.27	Input/Output (I/O) Count Limit Per Accounting User Class (UIOLIM) Macro	2-29
2.4.28	Processor Time Limit Per Accounting User Class (UPRCLIM) Macro	2-32
2.4.29	Multi-Terminal Monitor (MTM) Version Identification (VERSION) Macro	2-33
2.5	MEMORY REQUIREMENTS FOR SYSTEM GENERATION (SYSGEN) OPTIONS	2-34

3 AUTHORIZED USER UTILITY

3.1	INTRODUCTION	3-1
3.2	AUTHORIZED USER FILE (AUF)	3-1
3.3	BUILDING AND STARTING THE AUTHORIZED USER UTILITY	3-4
3.4	AUTHORIZED USER UTILITY COMMANDS	3-6
3.4.1	ACCOUNTS Command	3-8
3.4.2	ADD Command	3-10
3.4.3	CHANGE Command	3-12
3.4.4	DELETE Command	3-14
3.4.5	END Command	3-15
3.4.6	LIST Command	3-16

CHAPTERS (Continued)

3.4.7	MODE Command	3-18
3.4.8	PAUSE Command	3-19
3.4.9	PRIVILEGE Command	3-20
3.4.10	TOTAL Command	3-26
4	OS/32 ACCOUNTING FACILITY	
4.1	INTRODUCTION	4-1
4.2	SYSTEM REQUIREMENTS	4-2
4.2.1	OS/32 System Generation (Sysgen)	4-2
4.2.2	Multi-Terminal Monitor (MTM) System Generation (Sysgen)	4-2
4.2.3	Accounting Transaction File (ATF)	4-2
4.3	LINK OPTION	4-3
4.4	STARTING DATA COLLECTION	4-3
4.5	DATA COLLECTION	4-3
4.5.1	Accounting Date and Time Formats	4-3
4.5.2	Terminal Session or Batch Job Accounting Data	4-5
4.5.2.1	Elapsed Time	4-5
4.5.2.2	Signon and Signoff Date and Time	4-6
4.5.2.3	Job Number	4-6
4.5.2.4	Date and Time a Job is Submitted	4-6
4.5.2.5	Total Number of Tasks Executed	4-6
4.5.2.6	Total Number of Command Substitution System (CSS) Calls Executed	4-6
4.5.2.7	Total Number of Commands Entered	4-6
4.5.2.8	Total Number of DISPLAY FILES Commands Entered	4-7
4.5.2.9	User Identifier (Userid)	4-7
4.5.2.10	Renamed and Deleted Files	4-7
4.5.2.11	Account Number	4-7
4.5.3	Task-Related Accounting Data	4-7
4.5.3.1	Processor (PRC) Time	4-8
4.5.3.2	Roll Time	4-9
4.5.3.3	Wait Time	4-9
4.5.3.4	Date and Time a Task Started and Terminated	4-9
4.5.3.5	User Identifier (Userid)	4-9
4.5.3.6	Memory Used by a Task	4-10
4.5.3.7	Count of Input/Output (I/O) Transfers to I/O Devices	4-10
4.5.3.8	Roll Count	4-10
4.5.3.9	File Descriptor (fd) from Which a Task Was Loaded	4-11
4.5.3.10	User-Supplied Information	4-11
4.5.3.11	Account Number	4-11
4.5.3.12	Job Number	4-11
4.5.3.13	End of Task Code	4-11

CHAPTERS (Continued)

4.6	TERMINAL USER ACCESS TO ACCOUNTING INFORMATION	4-12
4.6.1	Task Termination	4-12
4.6.2	Task Control Block (TCB) Counter Overflow	4-12
4.6.3	Signoff Time	4-12
5	MULTI-TERMINAL MONITOR (MTM) OPERATING PROCEDURES	
5.1	INTRODUCTION	5-1
5.2	LOADING AND ESTABLISHING THE MULTI-TERMINAL MONITOR (MTM)	5-1
5.3	START COMMAND	5-2
5.4	.MTM COMMAND	5-4
5.5	MULTI-TERMINAL MONITOR (MTM) END OF TASK CODES	5-11
5.6	MULTI-TERMINAL MONITOR (MTM) ERROR MESSAGES	5-13
6	MULTI-TERMINAL MONITOR (MTM) OPERATION IN A MODEL 3200MPS SYSTEM MULTIPROCESSING ENVIRONMENT	
6.1	INTRODUCTION	6-1
6.2	OPERATION OF THE LOAD-LEVELING EXECUTIVE (LLE)	6-2
6.2.1	Queue 0 Eligibility	6-2
6.2.2	Prevention of Central Processing Unit (CPU) Overload	6-3
6.2.3	Error Termination	6-4
6.3	OPTIMIZATION AND CONTROL OF LOAD-LEVELING	6-4
6.4	LOAD-LEVELING OPTIMIZING AND CONTROL (.MTM LLE) COMMANDS	6-5
6.4.1	.MTM LLE Command	6-6
6.4.2	.MTM PSM Command	6-13
6.5	LOAD-LEVELING EXECUTIVE (LLE) OPERATING TIPS	6-15

APPENDIXES

A	MULTI-TERMINAL MONITOR (MTM) SYSTEM GENERATION (SYSGEN) OPTIONS	A-1
---	---	-----

APPENDIXES (Continued)

B	AUTHORIZED USER UTILITY COMMANDS	B-1	
C	AUTHORIZED USER UTILITY MESSAGE SUMMARY	C-1	
D	ACCOUNTING TRANSACTION FILE (ATF) RECORD FORMATS	D-1	
E	MULTI-TERMINAL MONITOR (^{.MTM} MTM) OPERATOR COMMAND SUMMARY	E-1	

FIGURES

1-1	MTM System Environment	1-1	
3-1	AUF Record Format	3-2	
5-1	Sample Output from the .MTM STATUS,USED Command	5-9	
D-1	ATF Format for Signon Account Record Type	D-2	
D-2	ATF Format for Signoff Account Record Type	D-3	
D-3	ATF Format for Task Completion (ATYP.TSK) and Task Data Overflow Account Record Types	D-4	
D-4	ATF Format for File Deletion Account Record Type	D-5	
D-5	ATF Format for File Renaming Account Record Type	D-6	

TABLES

2-1	BUFFER TYPES USED FOR SPECIFIC I/O REQUESTS	2-11	
2-2	MODIFICATION OF MTM SUBTASK PRIORITIES WITH PSM ENABLED	2-25	
2-3	EXAMPLE I/O COUNT LIMITS SET BY UIOLIM	2-31	
2-4	DATA STRUCTURES BUILT BY MTM SYSGEN	2-34	
3-1	AUTHORIZED USER UTILITY MODES AND ALLOWABLE COMMANDS	3-7	
3-2	MTM ACCOUNT PRIVILEGES	3-21	
5-1	MTM END OF TASK CODES	5-12	
6-1	LLE ERROR MESSAGES	6-10	
6-2	PSM ERROR MESSAGES	6-14	
D-1	ATF RECORD TYPES	D-1	

INDEX

IND-1 |

PREFACE

This manual is intended for system administrators and operators of an OS/32 Multi-Terminal Monitor (MTM) System. It provides step-by-step procedures for installing and operating an MTM system that is tailored to meet the needs of a particular organization.

Chapter 1 describes MTM and lists the system requirements and basic steps in configuring an MTM system. It also lists the recommended OS/32 system generation (sysgen) options that improve MTM efficiency. Chapter 2 outlines the MTM sysgen procedure and defines the MTM sysgen procedure options. Chapter 3 describes building, loading and starting the Authorized User Utility, and lists the Authorized User Utility commands. Chapter 4 discusses the OS/32 Accounting Facility option. Chapter 5 describes the operating procedures needed to load, start and control the MTM. Chapter 6 introduces the load-leveling executive (LLE) software and describes its use in enhancing MTM performance in a Model 3200MPS System multiprocessing environment. Appendix A summarizes the MTM sysgen options. Appendix B summarizes the Authorized User Utility commands. Appendix C lists the Authorized User Utility messages. Appendix D contains the accounting transaction file (ATF) record formats. Appendix E lists the MTM operator commands.

Revision 03 of this manual introduces LLE software and its associated commands. Chapter 6 describes the following commands to control and optimize LLE:

- .MTM LLE CPU
- .MTM LLE DISPLAY
- .MTM LLE ELIGIBILITY
- .MTM LLE LPU
- .MTM LLE OFF
- .MTM LLE ON

| Revision 03 includes the priority scheduling mechanism (PSM) and
| the description of options that enable the system operator to
| control the PSM.

| This manual is intended for use with the OS/32 R07.2 software
| release and higher.

For information on the contents of all Perkin-Elmer 32-bit
manuals, see the 32-Bit Systems User Documentation Summary.

CHAPTER 1
GENERAL DESCRIPTION

1.1 INTRODUCTION

The OS/32 Multi-Terminal Monitor (MTM) is an interactive multi-terminal, time-sharing system that executes as an executive task (e-task) in an OS/32 environment. MTM can be loaded, started and cancelled without disrupting the execution of other tasks running concurrently under OS/32. All tasks loaded and started from a user terminal execute as subtasks of MTM. Figure 1-1 shows the relationship of application tasks to OS/32 and MTM in a typical MTM system environment.

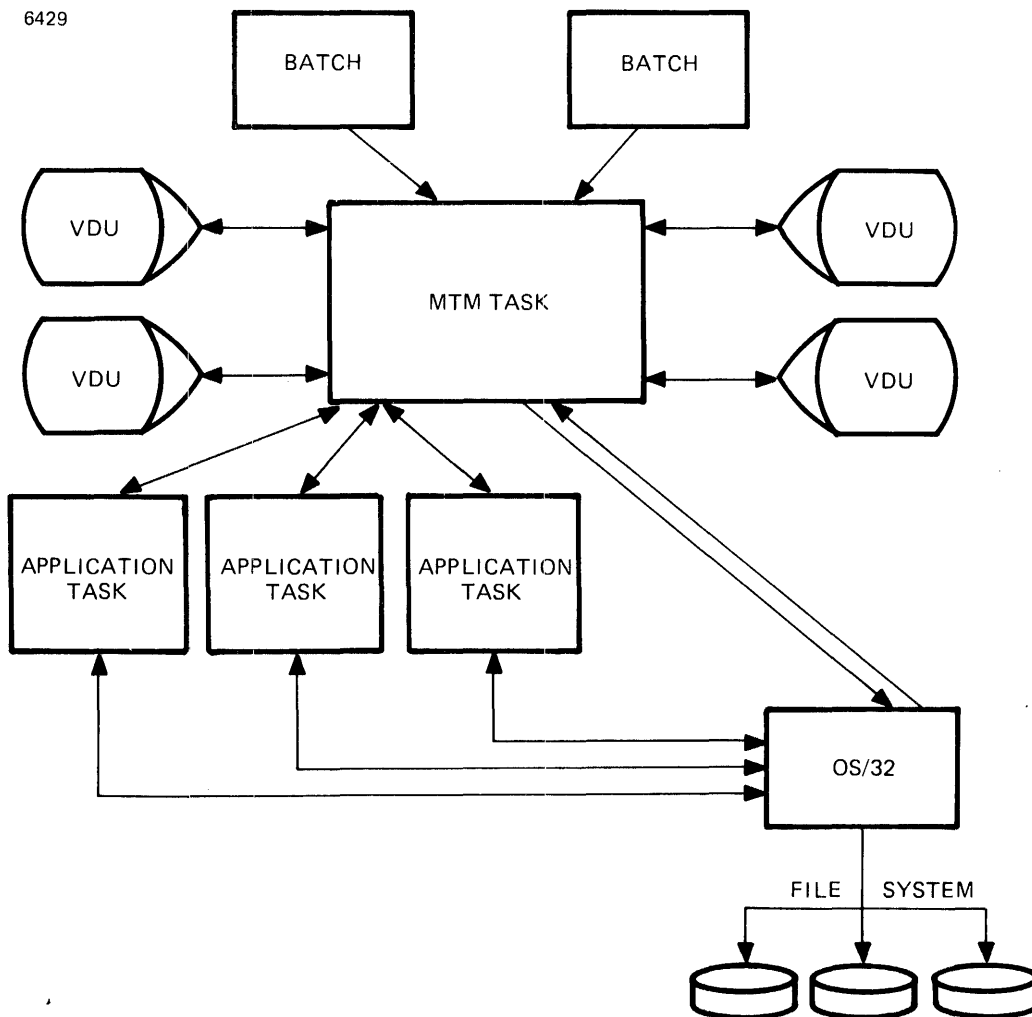


Figure 1-1 MTM System Environment

1.2 MULTI-TERMINAL MONITOR (MTM) SYSTEM REQUIREMENTS

The minimum hardware and software requirements for configuring an MTM system are:

- OS/32 R07.2 software release
- Perkin-Elmer Series 3200 Processor
- Memory access controller (MAC) or memory address translator (MAT), as appropriate
- Universal clock (UCLOCK) module
- Power fail/auto-restart option
- Console device
- Relocating (REL) loader, bootloader or loader storage unit (LSU) to load the operating system
- 256kb of memory (288kb or greater is recommended)
- Disk device
- ASCII terminal device

1.3 CONFIGURING A MULTI-TERMINAL MONITOR (MTM) SYSTEM

The following procedure is used to configure an MTM system.

1. Perform an OS/32 system generation (sysgen). See the OS/32 System Generation/32 (Sysgen/32) Reference Manual.
2. Unpack the MTM software. See the MTM packaging document for instructions.
3. Perform an MTM sysgen, modifying the MTM sysgen options to reflect installation requirements. See Chapter 2 for more information.
4. Build an authorized user file (AUF) using the Authorized User Utility described in Chapter 3.
5. Load and start MTM. See Chapter 5 for more information.

1.4 OS/32 SYSTEM GENERATION (SYSGEN) OPTIONS

When generating an operating system for an MTM system, select the following sysgen options to improve the overall efficiency and effectiveness of the system.

- Secondary directory support - adds a memory-resident directory to the system to reduce file search time.
- Segmentation support - allows the task image to be segmented into a private image area containing impure code and a shared image area containing pure code. During multiple simultaneous executions of the task, a copy of the private image is loaded into memory for each user, but only one copy of the shared image is loaded.
- Roll support - ensures that MTM can service a number of tasks in a limited amount of memory space.
- Spool support - provides input and output spooling capabilities to the MTM user.
- Accounting support - is required if the OS/32 Accounting Facility is to be used with MTM (see Chapter 4).

1.5 FILE DESCRIPTORS

Many of the command formats in this manual require the user to enter a file descriptor (fd). File descriptors are entered in the following format.

Format:

$$\left[\begin{array}{l} \{ \text{voln:} \} \\ \{ \text{dev:} \} \end{array} \right] [\text{filename}] [\text{.ext}] \left[/ \begin{array}{l} \{ \text{actno} \\ \{ \text{file class} \} \end{array} \right]$$

Parameters:

voln: is a 1- to 4-character disk volume name. The first character must be alphabetic and the remaining alphanumeric. If voln: is not specified, the default volume specified by the VOLUME command is used.

dev: is a 1- to 4-character device name. The first character must be alphabetic and the remaining alphanumeric.

filename is a 1- to 8-character name of a file. The first character must be alphabetic and the remaining alphanumeric. If a filename is specified when a device mnemonic is specified as voln:, the filename is ignored.

.ext is the name of the extension and consists of from 1- to 3-alphanumeric characters. For some commands, extensions are appended by default. If the period is omitted, the default is blanks.

actno is a decimal number ranging from 0 to 65,535 that specifies the account number associated with the file. Account numbers 1 through 65,535 are used by MTM. Account number 255 is a restricted account. MTM uses account 255 for the AUF. Account number 0 is used for system files and is the default for all operator commands entered from the system console. Tasks must have account privileges to specify an account number in this field.

file class is the class name of the file and consists of one character. The class names are:

- /P for private file
- /G for group file
- /S for system file

The file class can be specified by a terminal user or the system operator. If the system operator specifies /P, /G or /S, the operation is performed to account 0 only.

CHAPTER 2 MULTI-TERMINAL MONITOR (MTM) SYSTEM GENERATION (SYSGEN)

2.1 INTRODUCTION

This chapter describes MTM sysgen procedures and options available for configuring an MTM system.

The MTM task consists of two modules:

- MTMMAIN
- MTMSGN

MTMMAIN is a fixed length object file containing the executable code for MTM.

MTMSGN is a source file containing the MTM sysgen options. These options can be modified to meet installation requirements.

The procedure for performing an MTM sysgen is as follows:

- Select the MTM sysgen options that pertain to the installation.
- Use the command substitution system (CSS) command file (MTMSGN.CSS) to build the MTM task.

The following sections describe the MTM sysgen procedure in detail.

2.2 SELECTING MULTI-TERMINAL MONITOR (MTM) SYSTEM GENERATION (SYSGEN) OPTIONS

MTMSGN is a source module that contains the information necessary to configure an MTM system that meets the requirements of a particular installation. For example, one data field in this file tells MTM how many users can be signed on to the system concurrently. Another data field tells MTM how many command buffers to reserve in memory. See Section 2.4 for a complete description of the sysgen options contained in this file.

Perkin-Elmer supplies a standard MTM sysgen parameter file (MTMPARMS.MAC) with the MTM package. This file contains the following code:

```

**MTM.SGN
      SPACE 2
      VERSION
      SPACE 2
SGN.ACT EQU 0          NO ACCOUNTING SUPPORT
SGN.SCHR EQU 1        0=NONE / 1=DYNAMIC / 2=PSM
SGN.ROLL EQU 1        FORCE TASK ROLLABLE
SGN.PRIO EQU 6        BATCH PRIORITY
SGN.BTCP EQU 0        CANCEL BATCH JOB ON PAUSE
SGN.SPL EQU 0         NO SPL/32 SUPPORT
SGN.$HSP EQU 0        NO $HASP INTERFACE SUPPORT
SGN.MVOL EQU C'      NO DEFAULT USER VOLUME
SGN.VAR EQU 28        NUMBER OF VARIABLES
SGN.@GLO EQU 15       NUMBER OF GLOBAL VARIABLES
SGN.@INT EQU 8        NUMBER OF INTERNAL VARIABLES
NUSERS EQU 4          NUMBER OF USERS
NCSSLVS EQU 8         NUMBER OF CSS LEVELS
NCSBFS EQU 10        NUMBER OF EXTRA COMMAND BUFFERS
NCONBFS EQU 0         NUMBER OF CONSOLE BUFFERS
NSECTS EQU NUSERS/2  NUMBER OF SECTOR BUFFERS
NMSGs EQU NUSERS     NUMBER OF MESSAGE BUFFERS
QUESIZE EQU NUSERS   SIZE OF TASK QUEUE
CSBFSZ EQU 80        SIZE OF CSS BUFFERS
CONBFSZ EQU 0        SIZE OF CONSOLE BUFFERS
CDBUFSIZ EQU 3       SIZE OF SECTOR BUFFERS
PRCLIMIT EQU 0       NO PROCESSOR TIME LIMIT
      SPACE 2
      LINKOPT 4,30    LINK OPTIONS      IOB,LU

```

In addition, MTMSGN defines values for the prompt sequence, the input/output (I/O) class table, user I/O limits and user processor time limits. See the IOTAB, UIOLIM, UPRCLIM, PRMTSEQ and PRMTDEV macro instructions in Section 2.4 for more information on the default values for these sysgen options.

MTM sysgen options that differ from those defined by the standard MTM sysgen parameter file (MTMPARMS.MAC) can be selected by editing MTMPARMS.MAC via EDIT/32.

In the following example, MTMPARMS.MAC is edited to permit five accounting classes, SPL/32 and HASP support, assignment of a user default volume, 25 concurrent users, larger CSS buffers and an increased number of I/O blocks and logical units. In addition, several macro calls have been appended to MTMPARMS.MAC. Discussion of these macro calls is deferred until their respective sections in this chapter. Identification name '3230-B System' is assigned to this MTM system.

**MTM.SGN

```
SPACE 2
VERSION '3230-B SYSTEM'
SPACE 2
SGN.ACT EQU 5 5 ACCOUNTING CLASSES
SGN.SCHR EQU 1 0=NONE / 1=DYNAMIC / 2=PSM
SGN.ROLL EQU 1 FORCE TASK ROLLABLE
SGN.PRIO EQU 6 BATCH PRIORITY
SGN.BTCP EQU 0 CANCEL BATCH JOB ON PAUSE
SGN.SPL EQU 25 SPL/32 SUPPORT - 25 BUFFERS
SGN.$HSP EQU 1 1 $HASP INTERFACE
SGN.MVOL EQU C'MTM ' DEFAULT USER VOLUME - MTM
SGN.VAR EQU 28 NUMBER OF VARIABLES
SGN.@GLO EQU 15 NUMBER OF GLOBAL VARIABLES
SGN.@INT EQU 8 NUMBER OF INTERNAL VARIABLES
NUSERS EQU 25 NUMBER OF USERS
NCSLVS EQU 8 NUMBER OF CSS LEVELS
NCSBFS EQU 10 NUMBER OF EXTRA COMMAND BUFFERS
NCONBFS EQU 0 NUMBER OF CONSOLE BUFFERS
NSECTS EQU NUSERS/2 NUMBER OF SECTOR BUFFERS
NMSGs EQU NUSERS NUMBER OF MESSAGE BUFFERS
QUESIZE EQU NUSERS SIZE OF TASK QUEUE
CSBFSZ EQU 120 SIZE OF CSS BUFFERS
CONBFSZ EQU 0 SIZE OF CONSOLE BUFFERS
CDBUFSIZ EQU 3 SIZE OF SECTOR BUFFERS
PRCLIMIT EQU 0 NO PROCESSOR TIME LIMIT
SPACE 2
LINKOPT 25,80 LINK OPTIONS IOB,LU
SPACE 2
IOTAB (0,INDX),(1,CNTG),(2,DEV),(3,SPL),(4,TAPE)
SPACE 2
UPRCLIM 30,30,,,40,3600,80,10800
SPACE 2
UIOLIM (00,01,02,03,04),(100,101,102,103,104), U CLS 0 *
(10,11,12,13,14),, U CLS 1 *
(20,21,22,23,24),(120,121,122,123,124), U CLS 2 *
(30,31,32,33,34),(130,131,132,133,134) U CLS 3
SPACE 2
PRMTDEV (FOX,CT30,CT32,CT34), *
(SIGMA,SM50)
SPACE 2
PRMTSEQ (FOX,(*,08FF),(-,08FF),(>,FF),(BUILD>,FF)), *
(SIGMA)
```

It is recommended that the MTM sysgen parameter file that results from editing MTMPARMS.MAC be saved with a different extension. For example, MTMPARMS.MAC can be edited as above, with the new file saved as MTMPARMS.xxx where xxx is a 1- to 3-character alphanumeric extension. For a further explanation of this parameter, see Section 2.3.

See the OS/32 EDIT User Guide for information on editing files.

NOTE

The macro names VERSION, LINKOPT, IOTAB, UPRCLIM, PRMTDEV and PRMTSEQ must begin in column 10.

2.3 GENERATING A MULTI-TERMINAL MONITOR (MTM) SYSTEM USING THE STANDARD COMMAND SUBSTITUTION SYSTEM (CSS) COMMAND FILE (MTMSGN.CSS)

Perkin-Elmer supplies a CSS command file (MTMSGN.CSS) with the MTM package. MTMSGN.CSS assigns the SYSSTRUC.MLB and MTMSTRUC.MLB macro libraries. Therefore, MTMSGN.CSS may have to be edited via EDIT/32 to reflect the location of this file for a particular installation. For example, a volume and/or account classification (i.e., /P, /G, /S) may have to be specified for macro libraries. To call the standard CSS command file, enter the command as follows:

Format:

MTMSGN {xxx}
 {MAC}

Parameter:

xxx is a 1- to 3-character alphanumeric field that must be preceded by a space and must match the extension specified for the MTM sysgen parameter file (MTMPARMS).

Functional Details:

If the edited MTM sysgen parameter file was not saved with an extension different from that of the standard MTM sysgen parameter file or if the standard file is being used, MAC must be specified.

Successful execution of MTMSGN.CSS returns condition code = 0 upon completion of the MACRO/32 and CAL/32 processing phases. Upon successful completion of the link phase of MTMSGN.CSS, condition code = 1 is returned after the following message:

*** WARNING: ABSOLUTE SPACE LESS THAN 100 ***

The files created are a source listing MTMMAC.LST and a task MTMMAC.TSK. When a link error or assembly error (CAL/32) has occurred, see the source listing MTMMAC.LST. When a macro error (MACRO/32) has occurred, see macro expansion MTMMAC.EXP. If an extension was specified, see MTMxxx.LST, etc.

2.4 MULTI-TERMINAL MONITOR (MTM) SYSTEM GENERATION (SYSGEN) OPTIONS

The MTM sysgen options are:

- CDBUFSIZ - Sector buffer size
- CONBFSZ - Special console buffer size
- CSBFSZ - Command buffer size
- IOTAB - I/O class table
- LINKOPT - Number of I/O blocks and logical units for the Link OPTION command
- NCONBFS - Number of special console buffers
- NCSBFS - Number of command buffers
- NCSSLVS - Number of CSS levels
- NMSGs - Number of message buffers
- NSECTS - Number of sector length buffers
- NUSERS - Number of concurrent users
- PRCLIMIT - Batch job processor time limit
- PRMTDEV - MTM terminal prompt device name
- PRMTSEQ - MTM terminal prompt sequence
- QUESIZE - MTM task queue size
- SGN.ACT - Number of accounting classes
- SGN.BTCP - Batch job pause
- SGN.MVOL - Default user volume
- SGN.PRIO - Batch job relative priority
- SGN.ROLL - Forced roll
- SGN.SCHR - Scheduling mechanism

- SGN.SPL - Number of SPL/32 send data buffers
- SGN.VAR - Number of global and local variables
- SGN.@GLO - Number of new global variables
- SGN.@INT - Number of internal variables
- SGN.\$HSP - HASP interface
- UIOLIM - I/O count limit for each accounting user class
- UPRCLIM - Processor time limit for each accounting user class
- VERSION - MTM version identification

2.4.1 Sector Buffer Size (CDBUFSIZ) Option

The CDBUFSIZ option specifies the number of sectors in a sector length buffer. For example, using this option to increase the length of the buffers for the DISPLAY FILES and variable-related commands causes a corresponding increase in the speed of MTM. The standard MTMPARMS file specifies a value of 3 for this option. To modify the standard value, edit this option as follows.

Format:

CDBUFSIZ EQU n

Parameter:

n is a decimal number from 0 to 6.

Functional Details:

The size of each sector buffer is set according to the following formula:

$$(CDBUFSIZ * 256) + 256$$

The use of this option to increase the speed of the DISPLAY FILES command pertains only to files on disks that have been marked on with a secondary directory. The number of sectors calculated by the CDBUFSIZ option are read from the SYSTEM.DIR file each time a DISPLAY FILES command is executed.

If the disk is marked on with a secondary directory that can be contained in memory, MTM scans the memory buffer to search the directory. If CDBUFSIZ is set to 0, MTM does not scan the secondary directory to get the desired files; instead, the primary directory is scanned.

When variable support is included in an MTM system, the size and number of sector buffers allocated determine the number of variables that can be contained in memory. Sometimes a user will attempt to define more variables than can fit into the user's allocated sector buffers. When this occurs, one of the allocated sector buffers is paged to disk and a freed sector buffer is made available to the user. This allows the user to define more variables.

2.4.2 Special Console Buffer Size (CONBFSZ) Option

The CONBFSZ option specifies the maximum length of the special console buffers generated by the NCONBFS option. Since no special console buffers are reserved by the standard MTMPARMS file, the standard value of this option is 0. To specify the maximum length of each special console buffer reserved by MTM, edit this option as follows.

Format:

```
CONBFSZ EQU n
```

Parameter:

n is 0 or a decimal number from 72 to 32,767 specifying the length of each special console buffer in the system. If the standard value of this option is changed, the value of n should be greater than the number specified in the command buffer size (CSBFSZ) option (see Section 2.4.3).

2.4.3 Command Buffer Size (CSBFSZ) Option

The CSBFSZ option sets the maximum length of the MTM system command buffers, including the buffers used for CSS commands. The standard value for this option is 80. To modify the standard value, edit this option as follows.

Format:

```
CSBFSZ EQU n
```

Parameter:

n is a decimal number from 72 to 1,024.

Functional Details:

Since the last position of the command buffer is used for the carriage return (CR) character, the user should define a command buffer length one greater than needed. If a number less than 72 is specified for n, a message is output and MTM goes to end of task code 7 when the .MTM START command is entered.

2.4.4 Input/Output (I/O) Class Table (IOTAB) Macro

The IOTAB macro defines values for the I/O class table. To change the default values, enter the IOTAB macro instruction as follows.

Format:

```
[label] IOTAB (classno,classid [,iocount])0  
              ('...',  
              (classno,classid [,iocount])n
```

Operands:

classno	is a decimal number from 0 to 31 specifying a device or file class number.
classid	is a 4-character alphanumeric user-specified class identification associated with each device or file class. The classid is installation-dependent.
iocount	is a decimal number from 0 to 1,000,000 specifying the maximum number of I/O transfers to which each batch job associated with a particular device or file class is limited. If iocount is omitted, the default is 0 (no limit on I/O transfers).

Functional Details:

MTMSGN defines default values for the first four classes of the I/O class table as follows:

```
IOTAB (0,INDX,0),(1,CNTG,0),(2,DEV,0),(3,SPL,0)
```

These default values may be modified by appending the IOTAB macro call, with the desired parameters, to the MTMPARMS.xxx file. This is demonstrated in Section 2.3.

2.4.5 Multi-Terminal Monitor (MTM) Link (LINKOPT) Option

The LINKOPT option sets the maximum number of I/O blocks and logical units available to MTM. The standard MTMPARMS file permits four I/O control blocks and 30 logical units. To modify the standard values of LINKOPT, edit this option as follows.

Format:

LINKOPT n,m

Parameters:

n	is the maximum number of I/O control blocks available to MTM.
m	is the maximum number of logical units available to MTM.

Functional Details:

The number of I/O control blocks need only be four, even in a 64 user MTM environment.

The minimum number of logical units that should be specified can be determined by the following formula:

$$\text{logical unit (lu)} = 6 + (\text{NCSSLVS} + 1) * \text{NUSERS} \text{ or } 254, \\ \text{whichever is smaller.}$$

MTM dynamically allocates logical units to subtasks as needed. The actual number of logical units available to MTM for dynamic allocation can be calculated as follows:

$$\text{available logical units} = (\text{number of logical units specified at MTM sysgen (LINKOPT)}) - \text{NUSERS} - 6.$$

To avoid a decrease in response time, the recommended number of logical units for NUSERS greater than 5 is:

$$6 + \text{NUSERS} * 2 + (\text{NCSSLVS} * 2)$$

To determine if the number of logical units specified by LINKOPT is sufficient, check the number of times MTM goes into an lu wait state by using the .MTM STATUS, WAIT command. This command returns a table similar to the one shown below.

	.MTM ST,W					
13:14:32	.MTM: TOTAL	PEAK	CURRENT	WAITS		
13:14:34	.MTM: 42	34	11	0	CMD BUFFERS	
13:14:37	.MTM: 0	0	0	0	CON BUFFERS	
13:14:40	.MTM: 7	4	0	0	SEC BUFFERS	
13:14:42	.MTM: 50	24	19	0	LU	

A decision to increase or decrease the number of logical units specified by LINKOPT can be based on information returned by the .MTM STATUS, WAIT command.

The MTMMAIN object module has the following Link OPTION commands imbedded:

```
OPTION COMMUNICATION, CONTROL, XSVCl, ETASK, INTERCEPT, ABSOLUTE=0
```

```
| These options can be overridden by using the NDCMD command to
| suppress imbedded Link options or by explicit specification of
| desired options using Link commands.
```

The Link options IOBLOCKS and LU are contained in the MTMSGN source module. These options are selected as described above.

For more information on Link commands and options, see the OS/32 Link Reference Manual.

2.4.6 Number of Special Console Buffers (NCONBFS) Option

The NCONBFS option specifies the maximum number of special console buffers that can be used concurrently during MTM execution. No special console buffers are specified in the standard MTM sysgen parameter file. To specify the number of special console buffers to be reserved, edit this option as follows.

Format:

```
NCONBFS EQU n
```

Parameter:

n is a decimal number from 0 to 64.

Functional Details:

The maximum number of special console buffers that can be used at one time is equal to the number of users (NUSERS). These special console buffers are used only when an I/O request is made to a user terminal that requires an I/O buffer larger than the size specified in the command buffer size (CSBFSZ) option. An example of this is the use of graphic buffers.

Table 2-1 shows buffer types used in different situations.

TABLE 2-1 BUFFER TYPES USED FOR SPECIFIC I/O REQUESTS

I/O REQUEST SIZE	BUFFER TYPE	
	READ	WRITE
< CSBFSZ	TUB	TUB
> CSBFSZ and < CONBFSZ	CON	CON
> CSBFSZ and CONBFSZ = 0	TRUNCATE to CSBFSZ	DIRECT
> CONBFSZ	TRUNCATE to CONBFSZ	DIRECT

Where:

TUB (terminal user buffer) is the dedicated buffer of size CSBFSZ for each user.

CON (console) is the special console buffer of size CONBFSZ.

TRUNCATE truncates the read request.

DIRECT performs the write request directly from the task. The task is made nonrollable for the duration of the I/O call.

The decision to buffer large writes through the special console buffers or to perform the write directly from the task, should be based on:

- the speed of the terminal,
- the size of the write, and
- the impact on other system users. The user task (u-task) is nonrollable for the duration of the write request.

If an insufficient number of console buffers is allocated, a deadlock between two tasks might occur; e.g., both tasks enter into a console buffer wait state. If this happens, one of the two tasks involved should be cancelled. This deadlock situation is likely to occur if the special console buffers are used for read purposes and there is an insufficient amount of task space, causing tasks to be rolled. To monitor the number of console buffer waits, enter the .MTM STATUS parameter with the WAIT option.

2.4.7 Number of Command Buffers (NCSBFS) Option

The MTM program requires a minimum number of command buffers as follows:

- One buffer for each user (terminal or batch job)
- One buffer for each active CSS level
- One buffer for each MESSAGE command
- One buffer for each BROADCAST message
- One buffer for each new global or internal variable

MTM terminal response time depends on the number of available command buffers. To avoid poor performance, an adequate number of buffers must be allowed for MTM use. During MTM sysgen, three command buffers are reserved for each user; additional buffers can be reserved according to the number of CSS levels that may be active under MTM. The NCSBFS option tells MTM how many additional command buffers are to be reserved. The standard value for the NCSBFS option is 10. The length of these command buffers is specified by the CSBFSZ option.

To modify the standard value, edit this option as follows.

Format:

NCSBFS EQU n

Parameter:

n is a decimal number specifying the number of additional command buffers that can be used by the MTM system. Use the following formula to calculate n for your installation:

$$n = (\text{number of users}) \times (\text{number of CSS levels} + 1) / 3$$

The minimum number for n is 0.

Functional Details:

Response time may be adversely affected if the value specified for n is too small. Conversely, memory space is wasted if n is set above the maximum buffers needed.

The program development feature of MTM uses new global and new internal variables extensively. A user who plans to use the program development feature or expects to use a large number of new global and new internal variables, may require a large number of command buffers. In such cases, the value of NCSBFS should be increased beyond that of the calculated formula above. The recommended minimum number of extra command buffers that should be specified per user of the program development feature is 15.

The following methods can be used to judge whether the specified value for n is too small or too large for a particular system:

- Terminate MTM with the .MTM QUIESCE command. At end of task, MTM outputs the number of times it had to wait to get a command buffer.
- Enter .MTM STATUS, WAIT at the system console. This command displays the total number of buffers available, peak usage, current usage and the number of times MTM had to wait for system resources on the user's behalf.
- Enter .MTM STATUS, USED at the system console. This command displays in histogram form the number of times command buffers were used during MTM execution.

2.4.8 Number of Command Substitution System (CSS) Levels (NCSSLVS) Option

The NCSSLVS option sets the maximum nesting depth of CSS routines. If one CSS routine tries to call another and exceeds the designated depth, the system responds with an error message and CSS processing ceases. To modify the standard value, edit this option as follows.

Format:

NCSSLVS EQU n

Parameter:

n is a decimal number specifying the maximum number of CSS levels. The minimum value for n is one; the maximum value is 1,024.

2.4.9 Number of Message Buffers (NMSGs) Option

The NMSGs option defines the maximum number of message buffers that MTM can reserve for operator commands and accounting messages from the file manager. The standard value for this option is NUSERS. To modify the standard value, edit this option as follows.

Format:

NMSGs EQU n

Parameter:

n is a decimal number from 1 to 1,024. A recommended value for n is NUSERS.

Functional Details:

MTM reserves 76 bytes per buffer. If these buffers overflow, the following message is displayed by the operating system:

SVC6-ERR TYPE=NMSG POS=

2.4.10 Number of Sector Length Buffers (NSECTS) Option

The NSECTS option specifies the maximum number of sector length buffers that can be used by MTM for commands that require this buffer type. These commands are DISPLAY FILES, \$GLOBAL, \$LOCAL, \$SET and \$FREE. The number of sector length buffers specified by the standard MTMPARMS file is NUSERS/2. To modify the standard value, edit this option as follows.

Format:

NSECTS EQU n

Parameter:

n is a decimal number specifying the maximum number of 256-byte sector buffers. The minimum value for n is 1. A recommended value for n is NUSERS/2.

Functional Details:

When variable support is included in an MTM system through the SGN.VAR option, a sufficient amount of sector buffers should be reserved to hold MTM variables. A 256-byte (one sector) buffer can contain a maximum of seven variables. The amount of sector buffers needed depends upon the number of users, the number of variables being used, and the size of the sector buffers. Ideally, two 256-byte sector buffers per user should be allocated; a maximum of four per user can be allocated to a sector buffer pool. Very large sector buffers can also be allocated using the CDBUFSIZ option. If an insufficient number of sector buffers is allocated, the buffer pool can run empty, causing the user to go into a sector buffer wait state.

When variable information is not required, sector buffers are paged to disk. The resulting freed buffers are returned to the sector buffer pool. Because of this paging to and from disk, the size of each sector buffer is an important factor in processing variables.

2.4.11 Number of Concurrent Users (NUSERS) Option

The NUSERS option defines the maximum number of terminal users and batch jobs that can be executed concurrently. The value specified for this option in the standard MTMPARMS file is 4. To modify the standard value, edit this option as follows.

Format:

NUSERS EQU n

Parameter:

n is a decimal number from 1 to 64.

| 2.4.12 Batch Job Processor Time Limit (PRCLIMIT) Option

| The PRCLIMIT option defines the maximum processor (PRC) time (in
| seconds) for a batch job running under MTM. In a Model 3200MPS
| System environment, PRCLIMIT defines the maximum processor time
| (CPU+APU), for a batch job running under MTM. No processor time
| limit is set by the standard MTMPARMS file. To set the processor
| time limit, edit this option as follows.

Format:

| PRCLIMIT EQU n

Parameter:

n is a decimal number from 0 to 86,400 (24 hours) specifying the maximum processor time (in seconds) to which each batch job is limited. If 0 is specified, no limit applies.

2.4.13 Multi-Terminal Monitor (MTM) Terminal Prompt Device Name (PRMTDEV) Macro

The PRMTDEV macro defines the devices to be used with the prompt sequences specified in the PRMTSEQ macro. Any terminal device not defined by the PRMTDEV macro will use the standard prompt sequences. The PRMTDEV macro is used as follows.

Format:

```
[label] PRMTDEV (pname1, dev1 [, ..., devn])1  
                [, ..., (pnamen, dev1 [, ..., devn])n]
```

```
[label] PRMTDEV (SIGMA, dev1 [, ..., devn])
```

Operands:

pname is the 1- to 8-character prompt sequence name specified in the PRMTSEQ option (see Section 2.4.14).

dev is the device name of the terminal.

SIGMA is the predefined prompt sequence for the Perkin-Elmer SIGMA 10 terminal.

Example:

```
PRMTDEV (FOX, CT30, CT32, CT34), (SIGMA, SM50)
```

2.4.14 Multi-Terminal Monitor (MTM) Terminal Prompt Sequence (PRMTSEQ) Macro

The PRMTSEQ macro defines prompt sequences for terminals requiring special prompt sequences. This macro is used in conjunction with the MTM terminal prompt device name (PRMTDEV) macro and is only required if the user wishes to use a terminal prompt sequence other than the standard sequence.

Format:

[label] PRMTSEQ (SIGMA)

[label] PRMTSEQ (pname₁, (com-req,x), (task-exec,x),
(task-read,x), (build-read,x))₁
[, ...,
(pname_n, (com-req,x), (task-exec,x),
(task-read,x), (build-read,x))_n]

[label] PRMTSEQ (STANDARD, (com-req,x), (task-exec,x),
(task-read,x), (build-read,x))₁

Operands:

SIGMA	requests the predefined prompt sequence for the Perkin-Elmer SIGMA 10 terminal.
pname	is a 1- to 8-character alphanumeric prompt sequence name. This parameter corresponds to that specified for the PRMTDEV macro.
com req	is zero to eight ASCII characters defining the prompt sequence for a command request.
task exec	is zero to eight ASCII characters defining the prompt sequence for task or CSS execution.
task read	is zero to eight ASCII characters defining the prompt sequence for a task read.
build-read	is zero to eight ASCII characters defining the prompt sequence for a build read.
STANDARD	redefines the standard prompt sequence, which is as follows: <ul style="list-style-type: none">● Command read (*,FFFFFF)● Task/CSS executing (-,FFFFFF)● task read (>,FFFFFF)● Build read (B>,FFFFFF)
x	is a hexadecimal pad sequence from zero to eight bytes.

Functional Details:

The total length of the ASCII character sequence and the hexadecimal pad sequence cannot exceed eight bytes.

When specifying more than one set of prompt sequences, a comma must be placed immediately following the last parameter specified on a line and an asterisk (*) must appear in column 72 to designate continuation of the operands.

Example:

```
PRMTSEQ (BS,(*,08FF),(-,08FF),(>,FF),(BUILD>,FF)),          *  
(SIGMA)
```

2.4.15 Multi-Terminal Monitor (MTM) Task Queue Size (QUESIZE) Option

The QUESIZE option specifies the maximum size of the task queue used by MTM. The standard MTMPARMS file specifies NUSERS for this option. To modify the standard value, edit this option as follows.

Format:

```
QUESIZE EQU n
```

Parameter:

n is a decimal number specifying the maximum number of slots for the MTM task queue. The maximum value for this option is 200. A recommended value for n is NUSERS.

2.4.16 OS/32 Accounting Facility (SGN.ACT) Option

The SGN.ACT option enables support for the OS/32 Accounting Facility under MTM. The standard MTMPARMS file does not support the Accounting Facility, but to enable support and specify the maximum number of device and file classes that can be used, edit this option as follows.

Format:

```
SGN.ACT EQU n
```

Parameter:

n is a decimal number from 2 to 32 specifying the maximum number of device and file classes to be used by the Accounting Facility. A 0 indicates no accounting support.

Functional Details:

The value of n should be greater than, or equal to, the number specified in the OS/32 accounting configuration statement. A minimum value of 2 is required to support indexed and contiguous files.

2.4.17 Batch Job Pause (SGN.BTCP) Option

The SGN.BTCP option allows a batch job running under MTM to pause without being cancelled. The standard value of this option causes a batch job running under MTM to be cancelled whenever execution of that job is suspended. To enable the batch job pause option, edit this option as follows.

Format:

SGN.BTCP EQU 1

Functional Details:

The batch job pause option is enabled by equating SGN.BTCP to 1. If this option is enabled, a message is output to the system console when a batch job pauses. The system operator can then take appropriate action.

2.4.18 Default User Volume (SGN.MVOL) Option

The SGN.MVOL option defines the default volume for an MTM user. The standard MTMPARMS file defines the system volume as the default user volume. To specify a default volume, edit this option as follows.

Format:

SGN.MVOL EQU C'name'

Parameter:

name is the name of the disk volume to be used as the default user volume for all users at signon time. This name must be four characters; the first character must be alphabetic, the remaining alphanumeric. If less than four characters are used, the remaining characters must be blanks, e.g., C'MTM '. Four blank characters indicate that the default user volume is the system volume.

2.4.19 Batch Job Relative Priority (SGN.PRIO) Option

The SGN.PRIO option defines the relative default priority of a batch job running under MTM. To modify the standard value, edit this option as follows.

Format:

SGN.PRIO EQU n

Parameter:

n is a decimal number from 0 to 20 that is added to the MTM default priority (MTMdp) of an interactive job to determine the absolute priority at which an MTM user's batch job will run. For example, the standard value of 6 indicates that a batch job will run at six priorities lower than an interactive job unless otherwise specified. If 0 is specified, 0 is added to the MTMdp of an interactive job, indicating that a batch job will run at the same priority as an interactive job unless otherwise specified.

Functional Details:

Default priorities for interactive and batch jobs are calculated as follows:

TYPE OF JOB	PRIORITY
MTM	MTM link priority
Interactive	MTMdp = MTM link priority + 12
Batch	MTMdp + SGN.PRIO = MTM link priority + 12 + n

Once these default priorities are established, any priorities set by an MTM user in the SUBMIT command are checked against the default priorities. A batch job is always run at the lowest possible valid priority and is never run at the same priority as MTM. Default priorities for interactive jobs are not affected by this option.

Example:

If:

```
MTM link priority = 128
SGN.PRIO          = 10
```

the following become the MTM default priorities of interactive and batch jobs:

```
Interactive job = 140
Batch job      = 150
```

As shown in the following examples, priorities specified by an MTM user in the SUBMIT command are checked against the default priorities established at MTM sysgen.

Example:

Assume the following SUBMIT commands are entered by an MTM user (SGN.PRIO=10):

```
SUBMIT TEST.JOB,PR=138
SUBMIT TEST1.JOB,PR=150
```

The priority for TEST.JOB is invalid, because the user specified the submitted job to be run at a higher priority than the MTMdp of 140. The job will be run at 150.

The priority for TEST1.JOB is valid, because it is lower than the MTMdp of 139.

The batch job TEST1.JOB will be run at the specified priority of 150.

Example:

Assume the following SUBMIT commands are entered by an MTM user. The jobs will be run at either the MTMdp for batch jobs or the link priorities, whichever are lower (SGN.PRIO=6).

JOB	LINK PRIORITY
SUBMIT TEST.JOB	128
SUBMIT TEST1.JOB	200
SUBMIT TEST2.JOB,PRIO=160	180

TEST.JOB will be run at the batch priority of 146, which is lower than the link priority of 128.

TEST1.JOB will be run at the link priority of 200, which is lower than the batch priority of 146.

TEST2.JOB will be run at the link priority of 180, which is lower than both the submit priority of 160 and the batch priority of 146.

2.4.20 Forced Roll (SGN.ROLL) Option

In most operating system environments, it is desirable to make every task running under MTM rollable. With the SGN.ROLL option enabled, every task except executive tasks (e-tasks) loaded by MTM is made rollable at load time. The standard MTMPARMS file enables forced roll. To disable forced roll, edit this option as follows.

Format:

```
SGN.ROLL EQU 0
```

Functional Details:

This option applies only to the private image segment of the task. Shared segments are not rolled. See the OS/32 Application Level Programmer Reference Manual for a description of the task image segments established by Link.

2.4.21 Multi-Terminal Monitor (MTM) Scheduling (SGN.SCHR) Option

The user can select one of three scheduling techniques available through the SGN.SCHR option: OS/32 priority scheduling, dynamic priority scheduling and the MTM priority scheduling mechanism (PSM). The PSM is required for use of the load leveling executive (LLE) on a Model 3200MPS System (see Chapter 6). Dynamic priority scheduling is specified by the standard MTMPARMS file. To modify the standard value, edit this option as follows.

Format:

SGN.SCHR EQU n

Parameter:

n may be 0, 1 or 2 where:

- 0 indicates OS/32 priority scheduling.
- 1 indicates dynamic priority scheduling.
- 2 indicates the MTM PSM.

Functional Details:

When 0 is selected for n, no special task scheduling is done for MTM subtasks. The standard task scheduling of OS/32 is used; i.e., tasks are run according to their priority and the time-slice value from the SET SLICE command. Tasks of equal priority are scheduled on a first-in/first-out (FIFO) basis. See the OS/32 Application Level Programmer Reference Manual for more information on OS/32 priority scheduling.

When 1 is selected for n, dynamic priority and time-slice scheduling is activated for MTM subtasks. Through dynamic scheduling, a task's priority is adjusted through a range of 13 priorities, while a task's time-slice is adjusted through a range from 20ms to 2 minutes. Computation-intensive tasks (i.e., tasks that consume their time-slice without an interrupt) are adjusted to the next lower priority and their time-slice is doubled. I/O-intensive tasks (i.e., tasks that are interrupted before they consume their time-slice) are adjusted to the next higher priority and their time-slice is halved. Dynamic scheduling of MTM subtasks by OS/32 ensures that tasks requiring interactive terminal I/O are adjusted to the highest allowable priority.

When 2 is selected for n, the MTM PSM is activated for MTM subtasks. PSM increases system throughput by adjusting the priorities of MTM subtasks according to their run-time behavior. PSM performs a periodic check on the run-time behavior of all MTM subtasks and based on this behavioral information, PSM adjusts the priorities of subtasks whose initial load priorities had been in the range of *+12 to *+128, where * equals the task priority for .MTM. The interval of the above check is controlled by the operator command .MTM PSM INTERVAL described in Chapter 6.

All current subtask priorities outside this range and those priorities set by the PRIOR command or the SUBMIT command at load time are fixed. Fixed priorities cannot be modified by PSM and PSM cannot alter the maximum batch priority set by the SGN.PRIO option. Table 2-2 summarizes task priority modification when PSM is enabled.

PSM lowers the task priorities of subtasks requiring a relatively large amount of CPU time, while raising the priorities of subtasks that perform a large number of I/O operations. PSM will not raise a task's priority above the priority set at link time (except when the task is rolled).

All priorities assigned by PSM (except for rolled-out tasks) must be within the *+12 to *+128 range. To increase the chances of entry of a rolled-out subtask into the system, PSM gradually raises the priority of the subtask to *+1. Once the subtask is rolled-in, PSM adjusts its priority immediately within the *+12 to *+128 range.

Accounting support is required with PSM; both MTM accounting and OS/32 accounting must be sysgened for accounting support.

TABLE 2-2 MODIFICATION OF MTM SUBTASK PRIORITIES WITH PSM ENABLED

PRIORITY	DEFAULT TASKS	ALLOWABLE PRIORITY MODIFICATION
*	.MTM .MTMASST .SPL .SPLR	Fixed priority - cannot be modified by PSM.
* to +1 . . * to +11	None	Fixed priority range - can be modified by a PRIOR or SUBMIT command issued by a user with the privilege to alter batch priorities.
* to +12 * to +13	All MTM interactive tasks All batch MTM jobs	Subtasks whose priorities lie within this range can be modified by PSM according to their run-time behavior priorities for all tasks (except rolled-out tasks) adjusted by PSM must remain within this range.

TABLE 2-2 MODIFICATION OF MTM SUBTASK PRIORITIES WITH PSM ENABLED (Continued)

PRIORITY	DEFAULT TASKS	ALLOWABLE PRIORITY MODIFICATION
* to +13 (continued)		PSM raises or lowers task priorities for batch jobs as it does for interactive tasks, except that PSM cannot raise a batch job priority higher than the maximum set at MTM sysgen (see SGN.PRIO). NOTE Priorities in this range that have been set by the PRIOR or SUBMIT commands are boundaries for PSM. PSM only lowers these priorities as needed and never raises a priority higher than the priority set through the PRIOR or SUBMIT command.
* to +128 or 248 (whichever is smaller)		
* to +129 or 249 (whichever is smaller) . . . 254	None	Fixed priority range - can be modified by the PRIOR command or at link time, but only up to 254.

NOTE

To obtain good system performance, use the PRIOR 1 command in the standard system CSS (on Account 0, system volume) for interactive tasks; otherwise the tasks are lowered too far in priority.

Even a user without PRIOR command privileges is able to raise a task's priority if the PRIOR command is used in a CSS file on the system volume and system account.

See the Multi-Terminal Monitor (MTM) Reference Manual for more information on the PRIOR command.

2.4.22 SPL/32 Interface (SGN.SPL) Option

The SGN.SPL option gives MTM users access to the SPL/32 software. SPL/32 must be configured into the system during OS/32 sysgen; otherwise, this option has no effect on the system. The standard MTMPARMS file does not enable the SPL/32 interface. To enable SPL/32 support and specify the desired number of send data buffers, edit this option as follows.

Format:

```
SGN.SPL EQU n
```

Parameter

n is the number of send data buffers.

Functional Details:

When 0 is selected for n, SPL/32 support is disabled. If SPL/32 support is desired, a recommended value for n is NUSERS.

To judge whether the value specified for n is too small or too large for a particular system, enter .MTM STATUS, WAIT at the system console. This command displays the total number of buffers available, peak usage and current usage.

2.4.23 Global and Local Variables (SGN.VAR) Option

The SGN.VAR option specifies the maximum number of global and local variables that an MTM user can define. The maximum number specified by the standard MTMPARMS file is 28. To modify the standard value, edit this option as follows.

Format:

```
SGN.VAR EQU n
```

Parameter:

n is a decimal number from 28 to 1,024 specifying the combined maximum number of global and local variables a user can define. If n equals 0, variable support is not included in MTM.

2.4.24 Number of New Global Variables (SGN.@GLO) Option

The SGN.@GLO option specifies the maximum number of new global variables that can be defined by an MTM user via the \$DEFINE command. The maximum number of new global variables specified by the standard MTMPARMS file is 15. To modify the standard value, edit this option as follows.

Format:

SGN.@GLO EQU n

Parameter:

n is a decimal number from 0 to 99 specifying the maximum number of new global variables allowed.

Functional Details:

Each new global variable defined by a user occupies one command buffer. Therefore, the number specified for this option should be considered when specifying the NCSBFS option.

2.4.25 Number of New Internal Variables (SGN.@INT) Option

The SGN.@INT option specifies the maximum number of new internal variables that can be defined in an MTM user's CSS command file via the \$DEFINE command. The maximum number of new internal variables specified by the standard MTMPARMS file is eight. To modify the standard value, edit this option as follows.

Format:

SGN.@INT EQU n

Parameter:

n is a decimal number from 0 to 99 specifying the maximum number of new internal variables allowed in a CSS command file.

Functional Details:

Each new internal variable defined by a user occupies one command buffer. Therefore, the number specified for this option should be considered when specifying the NCSBFS option.

2.4.26 HASP Interface (SGN.\$HSP) Option

The SGN.\$HSP option allows terminal users to switch from an MTM time-sharing environment to a HASP environment. The standard MTMPARMS file does not enable the HASP interface. To enable the HASP interface, edit the SGN.\$HSP option as follows.

Format:

SGN.\$HSP EQU n

Parameter:

n is a decimal number from 1 to 99 specifying the maximum number of HASP/32 tasks that can be concurrently interfaced with the \$HASPxx command by MTM users. If n equals 0, the HASP interface feature is disabled.

2.4.27 Input/Output (I/O) Count Limit Per Accounting User Class (UIOLIM) Macro

The UIOLIM macro defines the maximum number of I/O transfers, within each I/O class, allowed for each task run from a terminal or batch stream, for a given user class (0 through 15). To define I/O count limits for each user class, first enable the OS/32 Accounting Facility using the SGN.ACT option and establish I/O classes for MTM using the IOTAB macro. Enter the UIOLIM macro as follows:

Format:

[label] UIOLIM (u₀io₀t, ..., u₀io_nt), (u₀io₀b, ..., u₀io_nb)
(u_nio₀t, ..., u_nio_nt), (u_nio₀b, ..., u_nio_nb)

Operands:

`uniont` is a decimal number ranging from 0 to 1,000,000 indicating the maximum number of I/O transfers to the device assigned to I/O class n for all interactive tasks run from a terminal for user class n. If `uniont` for a given user class is omitted or equals 0, the I/O count limits specified for each I/O class by the IOTAB macro are in effect for that class.

`unionb` is a decimal number ranging from 0 to 1,000,000 indicating the maximum number of I/O transfers to the device assigned to I/O class n for all batch jobs for user class n. If `unionb` for a given user class is omitted or equals 0, the I/O count limits specified for each I/O class by the IOTAB macro are in effect for that class.

Functional Details:

If a task attempts to exceed an I/O count limit set for the user class to which its account belongs, task execution is terminated with an end of task code 1210. All outstanding jobs corresponding to the exceeded I/O count limit are aborted.

Because the IOTAB macro defines the I/O classes for MTM, the IOTAB macro must precede UIOLIM in the MTM parameter file.

Interactive I/O counts for all I/O classes must be enclosed in parentheses; a comma is then used to separate the interactive I/O counts from the batch I/O counts (also enclosed in parentheses) for the same user class. Similarly, a comma following the batch I/O counts for all I/O classes is used to separate the parameters for one user class from those of the next user class.

The following example is designed to illustrate the format of the UIOLIM macro and its relationship to the IOTAB macro.

Example:

```

IOTAB (0,INDX),(1,CNTG),(2,DEV),(3,SPL),(4,TAPE)

UIOLIM (00,01,02,03,04),(100,101,102,103,104), U CLS 0 *
        (10,11,12,13,14),, U CLS 1 *
        (20,21,22,23,24),(120,121,122,123,124), U CLS 2 *
        (30,31,32,33,34),(130,131,132,133,134) U CLS 3 *

      I C D S T      I C D S T
      N N E P A      N N E P A
      D T V L P      D T V L P
      X G          E      X G          E
  
```

TERMINAL

BATCH

Table 2-3 illustrates the I/O count limits set for batch and terminal jobs by the above example of UIOLIM.

TABLE 2-3 EXAMPLE I/O COUNT LIMITS SET BY UIOLIM

ACCOUNTING USER CLASS	I/O CLASS									
	0 (INDX)		1 (CNTG)		2 (DEV)		3 (SPL)		4 (TAPE)	
	TERM.	BATCH	TERM.	BATCH	TERM.	BATCH	TERM.	BATCH	TERM.	BATCH
0	*	100	1	101	2	102	3	103	4	104
1	10	**	11	**	12	**	13	**	14	**
2	20	120	21	121	22	122	23	123	24	124
3	30	130	31	131	32	132	33	133	34	134
4	*	**	*	**	*	**	*	**	*	**

* No I/O count limit
 ** I/O count limit set by IOTAB macro

NOTES

The actual I/O count limits used in the previous example are for illustrative purposes only and should not be construed as recommended parameters to the UIOLIM macro.

| 2.4.28 Processor Time Limit Per Accounting User Class (UPRCLIM)
| Macro

| The UPRCLIM macro defines the maximum processor time allowed for
| each accounting user class for MTM tasks running in a single
| processor system. In a Model 3200MPS System, UPRCLIM defines the
| maximum processor time (CPU+APU) allowed for each accounting user
| class. There is a maximum of 16 accounting user classes (0
| through 15) established in the authorized user file (AUF). Each
| account is assigned to one of the established accounting user
| classes. A terminal session or batch job is limited to the
| processor time designated for the class to which the account
| belongs. To define processor time limits for each user class,
| first enable the OS/32 Accounting Facility using the SGN.ACT
| option. Then enter the UPRCLIM macro as follows.

Format:

| [label] UPRCLIM limit₀t, limit₀b, limit₁t, limit₁b,
|
| limit_nt, limit_nb

Operands:

limit_nt is a decimal number ranging from 0 to 86,400
(one day) indicating the maximum number of
seconds of processor time allowed any terminal
session for a user signed on to an account in
user class n. If limit_nt equals 0, no
processor time limit applies to terminal
sessions under class n.

limit_nb is a decimal number ranging from 0 to 86,400
(one day) indicating the maximum number of
seconds of processor time allowed any batch
job running under an account in user class n.
If limit_nb equals 0, the default processor
time limit is the limit set by the PRCLIMIT
option at MTM sysgen.

Functional Details:

| If execution of a task under any account exceeds the processor
| time limit set for the accounting user class to which the account
| belongs, the task is terminated (end of task code 1200). After
| termination, no other tasks can be loaded under that account and
| all batch jobs under that account are aborted.

Example:

```
UPRCLIM 30,30,,,40,3600,80,10800
```

In the above example, the processor time limit for both batch jobs and terminal users in accounting user class 0 is 30 seconds. Accounting user class 1 has no processor time limit for terminal users; batch jobs are restricted by the limit set in PRCLIMIT. For user class 2, the processor time limit for terminal users is 40 seconds, while batch jobs are limited to 3600 seconds (one hour). Class 3 users are limited to 80 seconds for terminal users and 10,800 seconds (three hours) for batch jobs. For the remaining classes, no processor time limits are set for terminal sessions; batch jobs running under these classes are limited to the value set by the PRCLIMIT option.

2.4.29 Multi-Terminal Monitor (MTM) Version Identification (VERSION) Macro

The VERSION macro assigns an identification name or number to the current revision of the MTM system. To specify an identification name for an MTMPARMS file, edit this option as follows.

Format:

```
[label] VERSION 'id'
```

Operand:

id is a 16-character alphanumeric string identifying the current revision of the MTM system. Special characters are allowed. If blanks are used, the identification name must be enclosed in quotes.

Example:

```
VERSION '3230-B'
```

2.5 MEMORY REQUIREMENTS FOR SYSTEM GENERATION (SYSGEN) OPTIONS

The MTM sysgen options reside in data structures built within the MTM task address space. Table 2-4 lists each data structure and its size.

TABLE 2-4 DATA STRUCTURES BUILT BY MTM SYSGEN

DATA STRUCTURE	MEANING	SIZE (IN BYTES)
Sysgen parameter area	Contains the sysgen parameter values to be used by the MTMMAIN module.	128
Terminal user block (TUB)	One is built for each concurrent MTM user defined by the NUSERS option.	$584 + (\text{NCSSLVS} + 1) * 8 + \text{CSBFSZ} + 4 + \text{SGN.ACT} * 4 + 4$
Command buffer area	Contains the command buffers.	$(3 * \text{NUSERS} + \text{NCBFS}) * (\text{CSBFSZ} + 4)$
Special console buffers	Contains the pool of special console buffers.	$\text{NCONBFS} * (\text{CONBFSZ} + 4)$
Message buffer area	Contains the message buffers.	$76 * \text{NMSG}$
Sector buffer area	Contains the sector buffers.	$(256 * \text{CDBUFSIZ} + 256) * \text{NSECTS}$
Circular list area 1	Contains the MTM task queue.	$4 * \text{QUESIZE} + 8$
Circular list area 2	Contains the command buffer wait queue.	$4 * \text{NUSERS} + 8$
Circular list area 3	Contains the console buffer wait queue.	$4 * \text{NUSERS} + 8$
Circular list area 4	Contains the sector buffer wait queue.	$4 * \text{NUSERS} + 8$

TABLE 2-4 DATA STRUCTURES BUILT BY MTM SYSGEN (Continued)

DATA STRUCTURE	MEANING	SIZE (IN BYTES)
Circular list area 5	Contains the lu wait queue.	4*NUSERS+8
Circular list area 6	Contains the message buffer wait queue.	4*NMSGs+8
Accounting write buffer	Used by MTM to write accounting information to the accounting transaction file (ATF).	128 or 8*classes+84 (whichever is larger)
Command buffer usage	Used to accumulate command buffer usage for the .MTM STATUS, USED command.	(3*NUSERS+NCSBFS)*4
I/O class table	Contains the I/O accounting class name and limits. This table is built by the IOTAB macro.	8*SGN.ACT
PRC limit table	Contains the PRC limits for terminals and batch jobs for each user class. Built by the UPRCLIM macro.	128
I/O limit table	Contains the I/O limits for terminals and batch jobs for each I/O class by user class. Built by the UIOLIM macro.	128*SGN.ACT
SPL/32 send data	Contains the mnemonic table, send data buffer queue and send data buffers used by MTM and SPL/32.	260*SGN.SPL+532

CHAPTER 3 AUTHORIZED USER UTILITY

3.1 INTRODUCTION

Prior to loading and starting the multi-terminal monitor (MTM), an installation must establish valid account numbers and passwords for its authorized users. This is accomplished through the Authorized User Utility supplied with the MTM software package. The Authorized User Utility maintains a file containing information on each account number. This file is called the authorized user file (AUF).

3.2 AUTHORIZED USER FILE (AUF)

The AUF contains one record for each account. An account number, group number and password are required for each account. In addition, user identification, limits for signon and processor time and privileges for each account can be optionally specified in the AUF. Defaults are set by the Authorized User Utility for all optional information not specified by the installation. The record format for the AUF is shown in Figure 3-1.

0(0)	Account number	12(2)	Group account number
4(4)			
8(8)	Password (12 characters)		
12(C)			
16(10)			
20(14)			
24(18)	Userid (20 characters)		
28(1C)			
32(20)			
36(24)	Signon time since last report		
40(28)	Reserved		
44(2C)	Date of last report		
48(30)			
52(34)	Total processor time		
56(38)	Signon time left (seconds)		
60(3C)	Processor time left (milliseconds)		
64(40)	Privileges		
68(44)			
72(48)	Reserved		
76(4C)			
80(50)			
84(54)	Filler		
88(58)			
120(78)	Filler		
124(7C)			

Figure 3-1 AUF Record Format

The Authorized User Utility is used to create, update and report on the AUF.

Initially, the Authorized User Utility should be run from the system console to create an AUF. In order to enable access under MTM, the AUF must be renamed to account 255.

Example:

```
RENAME USERS.AUF,USERS.AUF/255
```

Once the AUF is renamed to account 255, it can be updated while MTM is running by signing on to account 255 and running the Authorized User Utility.

When the AUF is assigned to account number 255, it is protected in two ways:

- only users signed on to account number 255 can access the AUF, and
- the Authorized User Utility does not display the password to account 255 in a listing.

When the LIST command of the Authorized User Utility is entered, the account number, group account number, user identifier (userid), signon time remaining, processor time remaining, account privileges and date of last report are displayed. Once added to an account record, the password cannot be retrieved from the AUF, but can be changed by the CHANGE command.

Accounting limits and privileges can be specified for each account number through one of the following Authorized User Utility commands:

- ADD command (create or update mode), used when adding a new account to the AUF.
- CHANGE command (update mode), used when making changes to an already existing account.

Two resources can be limited by account number:

- Signon time
- Processor time

Signon time can be limited for each account by specifying the maximum time the total number of terminal users assigned to that account can be signed on to the system. Whenever a terminal user signs off the system, these limits are checked. If the terminal user has used all of the remaining signon time specified for that account, or tries to sign on to an account with no signon time left, the signon request is rejected and this message is displayed:

NO TIME LEFT

Processor time can be limited for each account by specifying the maximum time the total number of terminal users assigned to that account can use time on a processor. In multiprocessing systems, processor time includes the total time used on the central processing unit (CPU) and any auxiliary processing unit (APU) assigned to the task. Whenever a terminal user's task terminates, this limit is checked. If the terminal user has used all of the remaining processor time specified for that account, this message is displayed:

YOUR PROCESSOR TIME HAS EXPIRED

If a terminal user tries to load a task assigned to an account with no processor time left, this message is displayed:

TIME-ERR

3.3 BUILDING AND STARTING THE AUTHORIZED USER UTILITY

The Authorized User Utility supplied with the package has been linked according to the sequence of Link commands shown below. User involvement is only necessary when additional Link options are required. In that case, the user must specify the necessary option and relink the task. See the OS/32 Link Reference Manual for more information.

```
L .BG,LINK
T .BG
ST
TITLE ACTUTY
ESTABLISH TASK
MAP ,ADDR
INCL ACTUTY.OBJ
BUILD ACTUTY
END
```

The following starts the Authorized User Utility.

Format:

$$\text{START, } \left\{ \begin{array}{l} \text{COMMAND} \\ \text{CMD} \end{array} \right\} = \text{fd}_1, \text{ LIST} = \text{fd}_2, \left\{ \begin{array}{l} \text{CREATE} \\ \text{UPDATE} \\ \text{REPORT} \end{array} \right\} = \text{fd}_3 \text{ [ACCOUNTS} = \text{n}]$$

Parameters:

COMMAND= file descriptor (fd₁) indicates the device or file from which command input will come. The Authorized User Utility assigns the command input device to logical unit 1 (lu1). The **COMMAND** parameter can also be entered as **C** or **CMD**.

LIST= fd₂ indicates the device or file to which the output is sent. The Authorized User Utility assigns the list device to lu3.

CREATE
UPDATE=
REPORT specifies a mode of operation for the Authorized User Utility. The AUF to be created, updated or about which a report is needed, is specified by fd₃. The Authorized User Utility assigns the specified AUF to lu3.

ACCOUNTS=n n is a decimal number (0 through 65,535) indicating the maximum number of accounts allowed in the AUF specified by fd₃. In the **CREATE** mode, n indicates the size of the new AUF. In the **UPDATE** mode, n indicates the size to which the AUF is to be extended. If the **ACCOUNTS** parameter is not specified, the maximum number of accounts in the AUF defaults to 255 or the maximum set by a previous **ACCOUNT** command. This parameter is ignored when the AUF is operating in the **REPORT** mode.

Functional Details:

When the **CREATE** mode is selected, the Authorized User Utility allocates, assigns and initializes the new file specified by fd₃. If the file specified by fd₃ already exists, an error message is generated and the Authorized User Utility goes to end of task. If the **UPDATE** mode is selected, an update to the file specified by fd₃ is requested. If the **REPORT** mode is selected, a report of the file specified by fd₃ is requested. If the specified file does not exist for the **UPDATE** and **REPORT** modes, an error message is generated and the utility goes to end of task.

The Authorized User Utility assigns all logical units based on the parameters specified in the START command. These starting parameters can be entered in any order. The user cannot assign specific logical units.

Examples:

The following example creates an AUF named NEWFILE. The system console (CON:) is the command input device and the line printer (PR:) is the list device. Until the START command is issued, the file (NEWFILE) does not exist. The Authorized User Utility allocates, assigns and initializes the new file for accounts 1 to 511.

```
START,CMD=CON:,LIST=PR:,CR=NEWFILE,AC=511
```

The next example specifies the console (CON:) as both the command input device and the list device. The mode of operation is UPDATE an existing file called AUFILF.

```
START,UP=AUFILF,CMD=CON:,LIST=CON:
```

3.4 AUTHORIZED USER UTILITY COMMANDS

The Authorized User Utility commands are:

- ACCOUNTS
- ADD
- CHANGE
- DELETE
- END
- LIST
- MODE
- PAUSE
- PRIVILEGE
- TOTAL

The Authorized User Utility has three modes of operation: create, report and update. Each mode allows certain Authorized User Utility commands to be executed (see Table 3-1).

- The create mode is used to establish a new AUF.
- The report mode allows an account record to be listed or an accounting report for each active account (an account record already assigned whether used or not) to be generated.
- The update mode allows account records to be added, deleted, modified or listed.

TABLE 3-1 AUTHORIZED USER UTILITY MODES AND ALLOWABLE COMMANDS

COMMANDS	MODES		
	CREATE	REPORT	UPDATE
ACCOUNTS	*	**	*
ADD	*		*
CHANGE	*		*
DELETE	*		*
END	*	*	*
LIST	*	*	*
MODE	*	*	*
PAUSE	*	*	*
PRIVILEGE	*	**	*
TOTAL		*	

* Command can be executed in specified mode

** Display (no operand) form only

ACCOUNTS

3.4.1 ACCOUNTS Command

The ACCOUNTS command raises or lowers the account number limit for the AUF. This command also displays the current limit set for the file.

Format:

ACCOUNTS n

Parameter:

n is a decimal number from 1 to 65,535 indicating the account number limit of the AUF. If n is omitted, the current account number limit is displayed. If the value specified for n is illegal, the ACCOUNTS command is ignored.

Functional Details:

In the REPORT mode, this command can be used in display form only. The Authorized User Utility sets the account limit in blocks of 256 records, one account per record. If n does not specify the last record in a 256 record block (e.g., 255, 1,023, etc.), the value of n is raised to the next multiple of 256. For example, if n=100, the account limit is set to 255.

After the ACCOUNTS command is executed, the following message is sent to the console:

ACCOUNT LIMIT = n

The variable n is the current maximum account number to which the AUF has been raised or lowered. If the ACCOUNTS command is entered without a value specified for n, this message will display the current account number limit in the AUF.

Examples:

ACCOUNT
ACCOUNT LIMIT = 2047

ACC 1000
ACCOUNT LIMIT = 1023

ADD

3.4.2 ADD Command

The ADD command establishes an account to the AUF.

Format:

```
ADD actno,groupno,password, [userid] [,{signontime}][,{processortime}]  
          [,privileges]
```

Parameters:

actno is a decimal number from 1 to the current account limit specifying the account number to be established.

groupno is a decimal number from 1 to the current limit (except 255) specifying a group account number to which the private account number is to be assigned. Group number 255 is valid only as a group number for account 255.

password is a 1- to 12-character alphanumeric string specifying the password to be established for the account.

userid is a 1- to 20-character alphanumeric string specifying the account user's identification.

signontime is a decimal number specifying the number of seconds of signon time to which the account is limited. If an asterisk (*) is specified, no limit is applied. For convenience, this parameter may be entered in any of the following formats:

- hh:mm:ss
- mm:ss
- seconds

For example, 1:30:10, 90:10 and 5410 all represent the same limit.

processortime is a decimal number specifying the processor time (CPU+APU) to which the account user is limited. If this parameter is omitted, the default established at MTM system generation (sysgen) is used. If an asterisk (*) is specified, no limit is applied. This parameter may be entered in the same formats as signontime.

privileges is a hexadecimal number specifying the privileges to be allowed for users signed on under this account. If this parameter is not specified, the default privileges 0000A800 (RENAME, ENVIRONMENT AT SIGNON and PRIVATE CSS EXECUTION) are assigned to the account. See Section 3.4.9 for information on account privileges and their corresponding hexadecimal codes.

Functional Details:

This command cannot be used in REPORT mode. The account added to the AUF must be inactive. If the account number is active, an error message is generated and the existing record is not changed. The Authorized User Utility mode must be CREATE or UPDATE. A copy of the record is written to the log device.

Account 255 must be added when the AUF is first created. A warning results if account number 255 does not exist when the Authorized User Utility is terminated.

The privileges for account 255 are set to FFFFFFFF regardless of the privileges specified when account 255 is established.

Example:

The following ADD command adds a new account to the AUF with the following information: the account number is 1000, the group number is 250, the password is RAY, the userid is DEPT 2, the signon time limit is 9999:59:59, and no processor time limit is set. The privileges in effect are SVC6 COMMUNICATION, CHANGE PASSWORD, PRIVATE CSS EXECUTION, TEMPORARY and SPOOL FILES, ENVIRONMENT AT SIGNON and RENAME.

```
ADD 1000,250,RAY,DEPT 2,9999:59:59,*,8000F800
    1000 250 DEPT 2 9999:59:59 ** 8000F800 03/03/83
```

CHANGE

3.4.3 CHANGE Command

The CHANGE command changes fields in the AUF record of an active account.

Format:

```
CHANGE actno, [groupno], [password], [userid] [ { signontime } ]
                                         [ * ]
|
[ { processortime } ] [ , privileges ]
  [ * ]
```

Parameters:

actno	is a decimal number from 1 to the current account limit specifying the account number to be changed.
groupno	is a decimal number from 1 to the current account limit (except 255) specifying the group number to which the account is to be assigned. Group number 255 is valid only as a group number for account 255.
password	is a 1- to 12-character alphanumeric string specifying the password for the account.
userid	is a 1- to 20-character alphanumeric string specifying the user's identification for the account.
signontime	is a decimal number specifying the additional signon time to which a terminal user is limited. If an asterisk (*) is specified, the currently established limit is changed and no limit is applied. For convenience, this parameter can be entered in any of the following formats:

- hh:mm:ss
- mm:ss
- seconds

For example, 1:30:10, 90:10 and 5410 all represent the same limit.

`processortime` is a decimal number specifying the additional processor time to which the terminal user is limited. If an asterisk (*) is specified, the currently established limit is changed and no limit is applied. This parameter can be entered in the same formats as `signontime`.

`privileges` is a hexadecimal number specifying the privileges to be allowed for users signed on under this account. See Section 3.4.9 for information on the account privileges and their corresponding hexadecimal codes. The privileges for account 255 default to FFFF.

Functional Details:

This command cannot be used in REPORT mode. As each record is changed, a copy of the original and resultant record is sent to the list device.

The values of `signontime` and `processortime` are added to the current `signontime` and `processortime` limits in the AUF to increase the time allocations. In a Model 3200MPS System, `processortime` is equal to the additional CPU+APU time to which the terminal user is limited. To decrease the time allocation, precede `signontime` or `processortime` with a minus sign (-). An asterisk (*) causes the limit to be disabled.

Example:

The following CHANGE command changes the processor time limit for account number 1000 from unlimited processor time to a limit of 1:30:10.

```
CH 1000,,,,,5410
 1000 250 DEPT 2          9999:59:59  **      8000F800 03/03/83
 1000 250 DEPT 2          9999:59:59  1:30:10  8000F800 03/03/83
```

DELETE

3.4.4 DELETE Command

The DELETE command makes an active account inactive.

Format:

DELETE actno

Parameter:

actno is a decimal number from 1 to the current account limit (except 255) specifying the account number to be made inactive.

Functional Details:

This command cannot be used in REPORT mode. A copy of the deleted record is written to the list device.

Examples:

The following DELETE command deletes account number 207.

```
DEL 207  
207 205 308503154TEMP SHOBER** ** A800 02/18/83
```

The following command attempts to delete an inactive account.

```
DEL 207  
*** ACCOUNT NUMBER 207 WAS INACTIVE
```

END

3.4.5 END Command

The END command closes all logical units and terminates the Authorized User Utility.

Format:

END

Functional Details:

If the Authorized User Utility is ended and account 255 does not exist, the Authorized User Utility issues a warning that the user has created an AUF that cannot be accessed later.

Example:

```
END
** WARNING ACCOUNT 255 DOES NOT EXIST
LAURIE -END OF TASK CODE= 0 PROCESSOR=9:43:33:04 TSK-ELAPSED=10:30:34 |
```

LIST

3.4.6 LIST Command

The LIST command displays active account records to the list device.

Format:

LIST [actno₁ [, actno₂]]

Parameters:

actno₁ is a decimal number from 1 to the current account limit specifying the account numbers to be displayed to the list device. If used in conjunction with actno₂, it specifies the beginning of a range of account numbers to be displayed to the list device.

actno₂ is a decimal number from 1 to the current account limit specifying the end of a range of account numbers to be displayed to the list device.

Functional Details:

When no account number is specified, all active accounts are listed. When only actno₁ is specified, only actno₁ is listed. When two account numbers are specified, all account numbers from the first account number to the second account number are listed.

Examples:

In the following example, a single account is listed.

LIST 1000

ACT	GRP NAME	TIME LEFT	PROCESSOR TIME LEFT	PRIV	DATE
1000	250 DEPT 2	9999:59:59	**	8000F800	03/03/83

A range of accounts is listed in the following example.

LIST 200,1000

ACT	GRP	NAME	TIME LEFT	PROCESSOR TIME LEFT	PRIV	DATE
201	205	308503154TEMP SHOBER	**	**	0000A800	02/18/83
207	205	308503154TEMP SHOBER	**	**	0000A800	02/18/83
211	205	308503154TEMP SHOBER	**	**	0000A800	02/18/83
230	230	035713087CAROL CARCH	**	**	0000A800	02/18/83
1000	250	DEPT 2	9999:59:59	1:30:10	8000F800	03/03/83

MODE

3.4.7 MODE Command

The MODE command displays the current mode, changes the current mode of the Authorized User Utility, or reassigns or creates an AUF.

Format:

$$\text{MODE } \left[\begin{array}{c} \text{(CREATE)} \\ \text{UPDATE} \\ \text{(REPORT)} \end{array} \right] [=fd_3]$$

Functional Details:

The MODE command with no parameters displays the current mode. When a mode is specified, the mode is reassigned but the AUF is not changed. When fd_3 is specified, the mode is changed and the AUF is reassigned.

When the create mode is specified, an error message is generated if fd_3 exists. The Authorized User Utility allocates the new AUF and initializes it for the number of account records currently in effect.

Examples:

To place the Authorized User Utility in the REPORT mode:

```
>MODE REPORT
```

To display the current mode:

```
>M  
***     REPORT IS THE CURRENT MODE
```

3.4.8 PAUSE Command

The PAUSE command pauses the Authorized User Utility and transfers control to the operating system. If the Authorized User Utility is running under MTM, control is transferred to MTM. The CONTINUE command returns control to the Authorized User Utility.

Format:

PAUSE

PRIVILEGE

3.4.9 PRIVILEGE Command

The PRIVILEGE command can be used to change or display the privileges for an account or range of accounts.

Format:

PRIVILEGE [actno₁ [actno₂] [{ privilege }]]

Parameters:

actno₁ is a decimal number from 1 to the current account limit specifying the account number or the start of a range of account numbers of which the current privileges are to be changed or displayed. If this parameter is omitted, the default is all currently active accounts.

actno₂ is a decimal number, from actno₁ + 1 to the current account limit, specifying the final account number of the range to be changed or displayed.

privilege is the hexadecimal mask of the privileges to be assigned to the specified account or range of accounts. If an asterisk (*) is specified, the default privilege 0000A800 is used. If this parameter is omitted, the current privileges of the specified account or range of accounts are displayed.

TABLE 3-2 MTM ACCOUNT PRIVILEGES

HEX MASK	PRIVILEGE AND DESCRIPTION	BIT
8000 0000	Supervisor call 6 (SVC6) Communication - allows the user to load a task with task communication enabled. If a user without this privilege loads a task with the SVC6 communication option, MTM disables the option when the task is loaded.	0
0800 0000	E-task - allows the user to load executive tasks (e-tasks). If a user without this privilege attempts to load an e-task, an error message is generated and the task is not loaded.	4
0400 0000	SVC6, logical processing unit (LPU) and APU Control - allows a user to load tasks with SVC6 control enabled. At load time, MTM checks the task's requested options against the user's privileges. If MTM discovers that the user does not have the SVC6 control privilege requested by the task, MTM disables the option at load time. If this option is set, the user may load a task with an LPU number other than 0 or may use the OPTIONS LPU=n and OPTIONS NLPU command. The user may load tasks with APC and APM options set (SVC13) at link time. If a user without this privilege loads a task, MTM resets the task LPU number to 0 (issuing a warning to the console) and disables the APC and APM options upon loading the task.	5
0080 0000	Extended Account Access - allows the user to enter an account number in addition to a class in an fd. This extended access is allowed for the commands RENAME, ASSIGN and LOAD, and for command substitution system (CSS) calls. File access is limited to shared read-only (SRO) for accounts other than the user's private account. Without this privilege, a user can only enter file classes in an fd; i.e., P, G, S.	8

TABLE 3-2 MTM ACCOUNT PRIVILEGES (Continued)

HEX MASK	PRIVILEGE AND DESCRIPTION	BIT
0040 0000	Set Group - allows the user to change the group account number of an account via the SET GROUP command. If a user without this privilege attempts to use the SET GROUP command, an error message is generated. The user's new group account number is retained until signoff or until changed by another SET GROUP command. The user's group number is not changed in the AUF. No user may set a group account to 255.	9
0008 0000	Task Account Privilege - allows the user to load tasks with the account privilege task option enabled. If a user without this privilege attempts to load a task requesting this option, MTM disables the option.	12
0004 0000	Bare Disk Access - allows the user to access bare disks for tasks that have this option enabled. Users requesting the bare disk option without this privilege will have the option disabled by MTM.	13
0002 0000	Universal Tasks - allows the user to run tasks as universal tasks. If a user without this privilege attempts to run a task with the universal option enabled, MTM disables the option and the task is run as a user-task (u-task).	14
0000 8000	Environment at Signon - permits the use of the ENV= parameter of the SIGNON command. If a user without this privilege tries to use the ENV= parameter, MTM ignores the environment request and forces the USERINIT.CSS to be executed.	16
0000 4000	Temporary and Spool Files - permit a user to display the filenames of temporary and spool files. If a user without this privilege attempts to display temporary or spool file fds, MTM returns *TEMPFILE* and *SPOOLFILE*.	17
0000 2000	Private CSS Execution - permits a user to execute private CSS files. A user without this privilege is limited to system CSS calls.	18

TABLE 3-2 MTM ACCOUNT PRIVILEGES (Continued)

HEX MASK	PRIVILEGE AND DESCRIPTION	BIT
0000 1000	Change Password - permits a user to select a new password via the PASSWORD command. The new password is entered in the AUF and remains in effect until changed by another PASSWORD command or the Authorized User Utility. An error message is generated for users without this privilege attempting to use the PASSWORD command.	19
0000 0800	Rename - permits a user to rename private files to group or system files; i.e., G, S. If a user also has the EXTENDED ACCOUNT ACCESS privilege, private files may be renamed to any account. Users without this privilege may rename files only within their private accounts.	20
0000 0400	Priorities 1 to 11 - permit a user to employ the PRIOR command in a private CSS to set a task's priority within the range of .MTM+1 to .MTM+11. The maximum priority available to users without this privilege is .MTM+12. Priorities .MTM+1 to .MTM+11 are always available to system CSS calls regardless of a user's privileges.	21
0000 0200	Batch Priority - allows a user to raise the priority of a batch job above the default priority. The default priority for batch tasks is .MTM+12+SGN.PRI0. Batch priorities can be raised through the use of the SUBMIT and PRIOR commands. The priority of a batch task can never be raised above the priority of interactive tasks (.MTM+12), but users may lower batch priorities regardless of their privileges.	22

TABLE 3-2 MTM ACCOUNT PRIVILEGES (Continued)

HEX MASK	PRIVILEGE AND DESCRIPTION	BIT
0000 0100	<p>Privileged User (Unlimited File Access) - allows a user unlimited access to all accounts except 255. Users with this privilege may enter the SET PRIVATE command to change their account numbers. This allows users to access the files of any account as if they were the user's private files; knowledge of the account's password is unnecessary. This privilege also allows users to change their group account numbers. The user retains all privileges from signon time when moving among accounts; the privileges of the account being accessed by a privileged user are ignored.</p> <p>This privilege also provides the user with the ability to display all batch jobs in the system through the INQUIRE command. Through the DISPLAY FILES command entered with an * in the account field, a user can display files in all accounts.</p>	23
0000 0080	<p>Foreground Interface - allows the use of the \$FOREGROUND interface. If a user without this privilege attempts to connect to a foreground task, an error message is generated.</p>	24
0000 0040	<p>HASP Interface - allows the user to interface with HASP tasks in the system. SGN.\$HSP must be enabled at MTM sysgen to permit this privilege. A user without this privilege who attempts to connect with a HASP task receives an error message.</p>	25
0000 000F	<p>Accounting User Classes - is used to specify an accounting user's class for each account. There are 15 (0 to F) possible accounting user classes. Each account must be assigned to an accounting user class. Each accounting class corresponds to a set of processor and input/output (I/O) limits as specified by the UPRCLIM and UIOLIM macros at MTM sysgen. If limits for each class are not established through UPRCLIM and UIOLIM, the limits set by PRCLIMIT and IOTAB apply, and all accounts effectively belong to the same accounting user class.</p>	28 to 31

Examples:

In the following example, the privileges of a single account are listed.

PRIV 1000			
ACT		NAME	PRIV
1000		DEPT 2	8000F800

In the following example, the privileges of a range of accounts are changed to the default privileges.

PRIV 200-1000,*			
ACT		NAME	PRIV
201		308503154TEMP SHOBER	0000A800
211		308503154TEMP SHOBER	0000A800
230		035713087CAROL CARCH	0000A800
1000		DEPT 2	0000A800

In the following example, the privileges of an account are changed.

PRIV 1000,FFFFFFF			
ACT	GRP	NAME	PRIV
1000	250	DEPT 2	FFFFFFF

TOTAL

3.4.10 TOTAL Command

The TOTAL command generates a report to the list device or resets the values in the signon time used and processor time used fields to 0.

Format:

TOTAL [C]

Parameter:

C is an alphabetic character specifying that the signon time and processor time used fields are to be cleared or reset to 0, and no report is to be generated to the list device. If this parameter is omitted, a report is generated to the list device and no values are cleared or reset to 0.

Functional Details:

The generated report lists each active account. The following details concerning each account are displayed: account number, group number, signon time left, processor time left, and the processor and signon time used since the last report was generated.

Example:

TOTAL

ACT	GRP	NAME	TIME LEFT	PROCESSOR TIME LEFT	PRIV	TIME USED	PROCESSOR TIME USED
1	1035213065	DCF GROUP	**	**	0000A800	0:00:00	0:00:00
10	18030503089	DC CONSULTA	**	**	0000A800	0:00:00	0:00:00
13	1035213065	SCHUCK PECK	**	**	8000FCOF	0:00:33	0:00:00
16	18031183089	BOB DOMITZ	**	**	8000FCOF	5:49:15	3:03:59
18	18031183089	WILL ETZ	**	**	8000FCOF	0:01:47	0:00:00
20	20035703085	OS32MT GROU	**	**	8COEFEOF	0:00:00	0:00:00
23	1035213065	CHRIS ARCHE	**	**	8000FCOF	0:00:00	0:00:00
29	18036243087	OS53	**	**	FFFFFFFF	0:00:00	0:00:00
33	18030493086	N SHKOLNIKO	**	**	8000FCOF	0:26:51	0:00:00
34	20032453088	ADMINISTRAT	**	**	8CDEFEOF	2:22:33	1:23:49
35	18032373086	GRANDEE HAFT	**	**	8000FCOF	0:00:00	0:00:00
36	20031543085	PETE HORN	**	**	8000FCOF	0:03:28	0:00:00
45	1035203065	JOHN GRAY	**	**	8000FCOF	0:00:00	0:00:00
53	96036243087	LANCE PATTI	**	**	8008FCOF	0:03:10	0:00:00
57	18031203089	SUE SELBY	**	**	8000FCOF	0:00:00	0:00:00
58	1035203065	BILL DEIBLE	**	**	8000FCOF	0:56:13	0:00:00

CHAPTER 4 OS/32 ACCOUNTING FACILITY

4.1 INTRODUCTION

The OS/32 Accounting Facility is a software feature that is optionally specified at OS/32 system generation (sysgen). This facility provides the user with management information for computer center operation. Information is collected on the basis of account numbers and user identifiers (userid), and is reported as a formatted summary on user-selected account numbers and time periods. The collected information includes such items as processor (PRC) time, auxiliary processing unit (APU) time, size of a task's segments, elapsed time, roll time, wait time, input/output (I/O) counts to I/O devices, and system space usage. Because data collection is performed by the operating system and the multi-terminal monitor (MTM), an operating system and MTM sysgen are required and should be consistent for both.

The Accounting Facility is composed of two parts:

- Data collection
- Reporting Utility

Data collection is performed by the operating system and MTM through counting, collecting and logging routines, which together collect and store the accounting information in a user-specified disk file called the accounting transaction file (ATF).

The Reporting Utility is a utility program that selects and transfers accounting information to a printer, disk file or tape file. Through operator-specified commands, accounting information can be used to produce cost distribution, billing, control and information reports. See the OS/32 Systems Support Utilities Reference Manual.

4.2 SYSTEM REQUIREMENTS

The Accounting Facility requires these system components:

- Perkin-Elmer 32-bit processor
- 256kb memory
- OS/32 R07.2 software release or higher
- Accounting support at OS/32 sysgen
- Accounting support at MTM sysgen
- Disk device
- Print device
- Temporary files

4.2.1 OS/32 System Generation (Sysgen)

The Accounting Facility is included in the OS by specifying the sysgen ACCOUNTING and IOCLASS configuration input commands. See the OS/32 System Generation (Sysgen/32) Reference Manual.

4.2.2 Multi-Terminal Monitor (MTM) System Generation (Sysgen)

The Accounting Facility is included in MTM through the parameter specified for the OS/32 Accounting Facility option (SGN.ACT) at MTM sysgen as described in Chapter 2.

4.2.3 Accounting Transaction File (ATF)

The ATF contains records for all activities performed by each account. The information processed by the Reporting Utility is provided by the ATF. The ATF is created by MTM through the ATF= parameter of the MTM START command (see Chapter 5). The ATF can also be changed by MTM through a parameter in the .MTM operator command. When the command .MTM ATF file descriptor (fd) is entered, the current ATF is closed and a new ATF is opened. This allows data collection to continue on the new ATF while the Reporting Utility processes the accounting data in the old ATF. If a new ATF is not opened, an assignment error is displayed and data can no longer be collected.

4.3 LINK OPTION

Accounting support can be enabled or disabled for tasks through an optional parameter in the Link OPTION command (see the OS/32 Link Reference Manual).

4.4 STARTING DATA COLLECTION

After operating system and MTM sysgens with accounting support have been performed, MTM is started as described in Sections 5.2 and 5.3. Data collection is started by specifying the ATF= parameter of MTM START command.

4.5 DATA COLLECTION

The OS/32 Accounting Facility collects data for each account. All collected data is stored in the appropriate ATF records on the basis of a job or task. Job-related data is collected on the total number of tasks submitted as one job. Task-related data is collected on a single task. Job- and task-related data are collected and stored in specific categories which are explained in the following sections.

See Appendix D for ATF record formats.

4.5.1 Accounting Date and Time Formats

Accounting dates are specified and displayed by MTM in these formats:

mm/dd/yy	is the month, day and year displayed on the terminal user's console when the terminal user signs off the system.
mon/dd/yy	is the month (three alphabetic characters), day and year specified in the Reporting Utility SELECT, OPTION and REPORT commands (see the OS/32 System Support Utilities Reference Manual).
dd/mon/yy	is the day, month (at least three alphabetic characters) and year specified in the Reporting Utility SELECT, OPTION and REPORT commands (see the OS/32 System Support Utilities Reference Manual).

Accounting times are displayed by MTM in these formats:

- a. hh:mm:ss is the hours, minutes and seconds of elapsed time, task elapsed time, signon time left or signoff time.
- b. hh:mm:ss.ms pp.p% is the hours, minutes, seconds, milliseconds (ms) and percentage of total elapsed time of user central processing unit (CPU) time, wait time, roll CPU time, supervisor call (SVC) CPU time, total processor time or total processor time left. In a Model 3200MPS System, the user APU time is displayed in this format.

The following displayable accounting times and their formats are specified:

- user CPU time is the CPU time spent directly by the user task; format b.
- SVC CPU time is the CPU time spent for SVCs on behalf of a user task; format b.
- roll CPU time is the CPU time spent to roll the user's task; format b.
- user APU time is the APU time spent by the user task (on Model 3200MPS only); format b.
- processor time is the total processor time spent by or on behalf of the user task (= user CPU + SVC CPU + roll CPU + user APU); format b.
- task elapsed time is the time between start and end of task; format a.
- signoff date and time is the date and time, in the appropriate format, of the terminal session or batch job signoff; format a.
- elapsed time is the time between signon and signoff for a terminal session or batch job; format a.
- total task elapsed time is the task elapsed times accumulated between signon and signoff for a terminal session or batch job; format a.
- total processor time is the processor time accumulated between signon and signoff for a terminal session or batch job; format b.

signon time left is the amount of time left for the terminal user or batch job out of the allocated limit; format a.

processor time left is the amount of processor time left for the terminal session or batch job out of the allocated limit; format b.

4.5.2 Terminal Session or Batch Job Accounting Data

The accounting data collected for each terminal session or batch job processed is:

- Elapsed time
- Signon and signoff date and time
- Job number
- Date and time a job is submitted
- Total number of tasks run
- Total number of command substitution system (CSS) calls executed
- Total number of commands entered
- Total number of DISPLAY FILES commands entered
- User identifier (userid)
- Renamed and deleted files
- Account number

All job-related accounting data is described in the following sections.

4.5.2.1 Elapsed Time

Elapsed time is the amount of time that passes between signon and signoff for a particular terminal session or batch job. Elapsed time is displayed as a message on the terminal user's console or batch log device each time the terminal user or batch job signs off:

ELAPSED TIME = elapsed time

4.5.2.2 Signon and Signoff Date and Time

The signon and signoff dates and times indicate the dates and times that the terminal user or batch job signs on and off the system. These dates and times are collected and stored as hexadecimal numbers in the ATF signon and signoff account records. Signon and signoff times are collected and stored in the ATF signon and signoff account records as the current time to the nearest minute.

The signoff date and time are displayed as a message on the terminal user's console or batch log device each time the terminal user or batch job signs off.

TIME OFF = signoff date and time

4.5.2.3 Job Number

The job number is a hexadecimal number assigned to a job consisting of one or more tasks. This number is collected and stored in the ATF signon, signoff and task completion or data overflow account records.

4.5.2.4 Date and Time a Job is Submitted

The date and time a job is submitted to the batch queue is collected and stored as a hexadecimal number in the ATF signon account record.

4.5.2.5 Total Number of Tasks Executed

The total number of tasks executed as part of a batch job or during a terminal session is collected and stored as a hexadecimal number in the ATF signoff account record.

4.5.2.6 Total Number of Command Substitution System (CSS) Calls Executed

The total number of CSS calls executed by a batch job or during a terminal session is collected and stored as a hexadecimal number in the ATF signoff account record.

4.5.2.7 Total Number of Commands Entered

The total number of commands entered in the system by a batch job or during a terminal session is collected and stored as a hexadecimal number in the ATF signoff account record.

4.5.2.8 Total Number of DISPLAY FILES Commands Entered

The total number of DISPLAY FILES commands entered in the system by a batch job or during a terminal session is collected and stored as a hexadecimal number in the ATF signoff account record.

4.5.2.9 User Identifier (Userid)

The userid, with which the terminal user or batch job signs on the system, is collected and stored in the ATF signon, signoff and task completion or data overflow account records.

4.5.2.10 Renamed and Deleted Files

Data pertaining to files that are renamed or deleted is collected and stored in the ATF renaming or deletion account records. This data consists of the:

- Date and time the file was created
- Current fd
- Number of sectors in the file
- Date and time the file was renamed
- New fd
- New account number to which the file was assigned

4.5.2.11 Account Number

The account number with which the terminal user or batch job signs on the system is collected and stored in all the ATF account records.

4.5.3 Task-Related Accounting Data

The accounting data collected for each task processed is:

- User CPU time, SVC CPU time and roll CPU time
- User APU time (Model 3200MPS System only)
- Roll time
- Wait time
- Date and time a task started and terminated

- Userid
- Memory used
- Count of I/O transfers to I/O devices
- Roll count
- fd from which a task was loaded
- User-supplied information
- Account number
- Job number
- End of task code

All task-related accounting data is described in the following sections.

4.5.3.1 Processor (PRC) Time

Processor time is collected and stored in the ATF task completion or data overflow account records as a number of 100ms increments. All processor time is collected and stored in the following categories:

- User CPU time
- | ● SVC CPU time
- Roll CPU time
- | ● User APU time

User CPU time is the amount of CPU time spent executing user code and nonreentrant system code. SVC CPU time is the amount of CPU time spent executing reentrant system code on behalf of the user. Roll CPU time is the amount of CPU time spent rolling a task into memory. User APU time is the time spent executing user code on an APU in a Model 3200MPS System. Processor time (user CPU + user APU + SVC CPU + roll CPU) is displayed as a message on the terminal user's console or batch log device. The following message is displayed when a terminal user's task or batch job terminates:

```
| taskid-END OF TASK CODE=xxxx PROCESSOR=processor time TSK-ELAPSED=task elapsed time
```

The following message is displayed when data overflow occurs:

```
taskid-ACCOUNTING DATA RESET PROCESSOR=processor time TSK-ELAPSED=task elapsed time |
```

The following message is displayed when the terminal user or batch job signs off:

```
PROCESSOR=total processor time TSK-ELAPSED=total task elapsed time |
```

The user CPU, SVC CPU, roll CPU and user APU times can also be displayed by the DISPLAY ACCOUNTING command (see the OS/32 Multi-Terminal Monitor (MTM) Reference Manual).

4.5.3.2 Roll Time

Roll time is the time a task spends waiting to execute because it was rolled out of memory. All roll time is collected and stored in the ATF task completion or data overflow account records as a number of 100us increments. Roll time can be displayed by the DISPLAY ACCOUNTING command (see the OS/32 Multi-Terminal Monitor (MTM) Reference Manual).

4.5.3.3 Wait Time

Wait time is the time a task spends waiting to execute because of a request that task issued for itself (e.g., I/O request). Wait time excludes all PAUSE requests and the time a task spends suspended by another task or by the terminal user. All wait time is stored in the ATF task completion or data overflow account records as a number of 100ms increments. Wait time can be displayed by the DISPLAY ACCOUNTING command (see the OS/32 Multi-Terminal Monitor (MTM) Reference Manual).

4.5.3.4 Date and Time a Task Started and Terminated

The date and time a task started and terminated is collected and stored in the ATF task completion or data overflow account records as a number of 100ms increments.

4.5.3.5 User Identifier (Userid)

The userid, with which the terminal user or batch job signs on the system, is collected and stored in the ATF signon, signoff and task completion or data overflow account records.

4.5.3.6 Memory Used by a Task

The total amount of memory used by a task during execution is collected and stored in the ATF task completion or data overflow account records as the number of 256-byte segments. Memory used is collected and stored in three categories:

- Impure segment space
- Shared segment space
- System space

Impure segment space is the maximum amount of impure memory used by a task during execution. Shared segment space is the maximum amount of shared memory (i.e., pure, TCOM, RTL) used by a task during execution, and system space is the maximum amount of system space used by a task during execution.

4.5.3.7 Count of Input/Output (I/O) Transfers to I/O Devices

All I/O transfers (read, write, test and set) made to an I/O device assigned to a particular device class are counted, collected and stored as a hexadecimal number in the ATF task completion or data overflow account records. These numbers are stored in the appropriate class category in the ATF. Each I/O class supported by the Accounting Facility occupies eight bytes in each ATF account record. The number of bytes transferred every time an I/O transfer occurs is counted and stored as a hexadecimal number in the ATF. The count of I/O transfers can be displayed by the DISPLAY ACCOUNTING command. I/O transfer limits are established at MTM sysgen time by the IOTAB or UIOLIM macros (see Chapter 2) and can be overridden by the terminal user for batch jobs through the SIGNON command (see the OS/32 Multi-Terminal Monitor (MTM) Reference Manual).

4.5.3.8 Roll Count

Roll count is the total number of times the impure segment of a task is rolled into memory. The roll count is collected and stored as a hexadecimal number in the ATF task completion or data overflow account records. The roll count can be displayed by the DISPLAY ACCOUNTING command (see the OS/32 Multi-Terminal Monitor (MTM) Reference Manual).

4.5.3.9 File Descriptor (fd) from Which a Task Was Loaded

Every time a task is loaded, the fd from which it was loaded is stored in the ATF task completion or data overflow account records. This data aids the user in determining which system tasks are used more often, and whether they should be made resident or nonresident.

4.5.3.10 User-Supplied Information

User-supplied information is eight bytes of data supplied by the terminal user through the SETACCT macro. This information is stored in the ATF task completion or data overflow account records (see the OS/32 System Macro Library Reference Manual).

4.5.3.11 Account Number

The account number, with which the terminal user or batch job signs onto the system, is collected and stored in all ATF account records.

4.5.3.12 Job Number

The job number, to which a particular task is assigned, is collected and stored in the ATF signon, signoff and task completion or data overflow account records.

4.5.3.13 End of Task Code

The end of task code, with which each task terminates, is collected and stored in the ATF task completion or data overflow account records. The code is displayed on the terminal user's console or batch log device each time a user's task or batch job terminates:

```
taskid-END OF TASK CODE=xxxx PROCESSOR=processor time TSK-ELAPSED=task elapsed time ;
```

If a task exceeds the processor time limit, the end of task code 1200 is displayed when the task terminates. If a task exceeds the I/O transfer limit, the end of task code 1210 is displayed when the task terminates.

4.6 TERMINAL USER ACCESS TO ACCOUNTING INFORMATION

A terminal user can access accounting information:

- at task termination,
- when a task control block (TCB) counter overflow occurs,
- at signoff time,
- through the DISPLAY ACCOUNTING command (see the OS/32 Multi-Terminal Monitor (MTM) Reference Manual), or
- through the fetch accounting information (FETACCT) macro (see the OS/32 Multi-Terminal Monitor (MTM) Reference Manual).

4.6.1 Task Termination

When a terminal user's task or batch job terminates execution, accounting information is displayed on the terminal user's console or batch log device in this message:

```
taskid-END OF TASK CODE=xxxx PROCESSOR=processor time TSK-ELAPSED=task elapsed time
```

4.6.2 Task Control Block (TCB) Counter Overflow

When the counters for data collection in the TCB overflow, this accounting information is displayed on the terminal user's console or batch log device in this message:

```
taskid-ACCOUNTING DATA RESET PROCESSOR=processor time TSK-ELAPSED=task elapsed time
```

4.6.3 Signoff Time

When a terminal user or batch job signs off the system, accounting information is displayed to the terminal user's console or batch log device in these messages:

```
ELAPSED TIME=elapsed time      PROCESSOR=total processor time TSK-ELAPSED-total task elapsed time  
SIGNON LEFT=signon time left   PROCESSOR LEFT=processor time left  
TIME OFF=signoff date and time
```

If there are no signon or processor time limits set for the account, SIGNON LEFT= and PROCESSOR LEFT= are not displayed at signoff.

CHAPTER 5
MULTI-TERMINAL MONITOR (MTM) OPERATING PROCEDURES

5.1 INTRODUCTION

Before loading and starting MTM, an authorized user file (AUF) must already exist (see Chapter 3). The commands used to load, establish, start and control MTM are presented in this chapter.

5.2 LOADING AND ESTABLISHING THE MULTI-TERMINAL MONITOR (MTM)

The following two commands load and establish MTM as the current task:

Formats:

```
LOAD .MTM,fd
```

```
TASK .MTM
```

Parameter:

fd is the file descriptor of the device from which MTM is loaded.

Functional Details:

The priority of MTM is established when MTM is linked. To ensure that MTM is not at the same level as the real-time system, MTM should be set at a priority lower than the lowest priority of real-time tasks in the operating system. If MTM is set at a lower priority, it executes as a background system. Since MTM is almost totally input/output (I/O) bound, it does not interfere with other tasks. Tasks executing in an MTM environment will run at a maximum priority of one less than the priority of MTM.

START

ST, AUF=USERS, AUF/255

5.3 START COMMAND

The START command starts MTM. Parameters for the START command may be specified in any order.

Format:

```
START ,AUF=fd [,ATF=fd] [,TERMINAL=(fd1 [,fd2, ...,fdn])]
      [,BATCH=(n,fd)] [,PAGE=fd]
```

Parameters:

AUF= fd is the file descriptor of the AUF. This file must reside on account 255. The Authorized User Utility must be used to create the AUF prior to running MTM. This parameter is required.

ATF= fd is the file descriptor of the accounting transaction file (ATF). The ATF accumulates accounting data in systems with accounting support. If the indicated file exists, records are appended. If it does not exist, MTM automatically allocates it. This parameter must be specified if the operating system is generated with accounting support. A user who does not wish to retain accounting information can specify ATF=NULL.

TERMINAL= fd is the file descriptor of the terminal devices that MTM is to monitor. Up to 64 terminal fds can be specified. If the command line entered exceeds one line, specify additional terminal devices with the .MTM ADD command.

BATCH= n is a decimal number from 1 to 64 indicating the number of concurrent batch streams that MTM can start.

fd is the file descriptor of the batch queue. The batch queue is an indexed file containing fds of the batch streams to be processed. If the fd specified does not exist, MTM allocates the file.

PAGE-

fd is the file descriptor of the page file used by local and global variables. If this parameter is omitted, MTM automatically allocates a file named PAGE on the system volume, account 255.

Functional Details:

The number of concurrent batch streams allowed, plus the number of terminals to be monitored, cannot exceed the maximum number of users allowed (NUSERS) at MTM system generation (sysgen).

MTM can optionally load and start an assistant task (which runs under the task name .MTMASST). An assistant task executes a supervisor call 7 (SVC7) on behalf of MTM. This task can improve MTM response time because MTM does not have to wait while an SVC7 is in progress.

When started, MTM tries to load and start the assistant task. MTM assumes that the assistant task image is on the file MTMASST.TSK on the system volume. If this file does not exist, or if the load fails, MTM is not affected. However, MTM issues its own SVC7 calls and response time is slower.

Because the operating system cancels all subtasks when the monitor task goes to end of task, .MTMASST is also cancelled when MTM goes to end of task. If an attempt is made to restart MTM, MTM goes to end of task 254.

When MTM is started from a command substitution system (CSS) file, and an .MTM command immediately follows the START command, an SVC6-ERR TYPE=QUE POS=.MTM error may occur. This error results if the MTM task has not completed initialization when the .MTM command is entered. To avoid this problem, put time delay commands such as \$WAIT 2 between the START and .MTM commands.

Example:

This command starts MTM using an AUF named ACCTS.32/255 and an ATF named ATF./255. Three terminals are to be monitored. Four batch jobs may run concurrently, and the file BATCH.Q is assigned as the batch queue.

```
ST,AUF=ACCTS.32,ATF=ATF,T=(CR10:,CR12:,CR14:),B=(4,BATCH.Q)
```

.MTM

5.4 .MTM COMMAND

The .MTM command enables the system operator to control MTM, its terminals and users from the system console.

Format:

|

.MTM	STATUS	{	WAIT	}
			TERMINAL	
			BATCH	
			USED [,fd ₁]	
			current batch and terminal	
	{	TELL	}userid message	
		MESSAGE		
		BROADCAST message		
	ADD	{	device:	}
			BATCH n [,fd]	
		TERMINATE device:		
	REMOVE	{	device:	}
			BATCH n	
		QUIESCE		
		PURGE fd		
	INQUIRE [fd]			
	SUBMIT fd			
	ATE fd			

Parameters:

STATUS displays current information. If no further parameter is specified, information on terminal users and batch jobs is displayed.

WAIT displays to the system console system resources available to MTM, the peak usage of system resources, and the number of times users had to wait for resources to become available.

TERMINAL indicates that terminal information is to be displayed.

BATCH indicates that batch information is to be displayed. The number of batch streams that can run concurrently is also displayed.

USED displays in histogram form the number of times a command buffer was used during MTM execution. See Figure 5-1 for an example of a listing output by the USED parameter.

fd₁ specifies the file or device to which the output from the USED parameter is sent. If fd₁ is omitted, the system console is the display device.

TELL displays a message on a specified user terminal.

MESSAGE displays a message on a specified user terminal.

userid identifies the terminal user.

BROADCAST displays a message on all terminals whether or not they are in use.

message is a 1- to 62-character alphanumeric string indicating the message to be broadcast.

ADD adds an additional terminal or concurrent batch to the MTM system.

device is the new terminal to be added. The total number of terminals supported by MTM must not exceed NUSERS (number of batch streams).

BATCH n is the number of additional concurrent batch streams to be allowed. The total number of concurrent batch streams supported by MTM must not exceed NUSERS (number of terminals). This parameter increases the number of allowable concurrent batch streams supported by MTM but does not add batch streams to the batch queue. To add batch streams to the batch queue, use the SUBMIT command.

fd specifies the file descriptor of the batch queue file. If a batch queue currently exists, this parameter is ignored.

TERMINATE immediately cancels a terminal user's task and signs off a terminal user. This parameter does not affect batch jobs.

device is the terminal to which the user to be terminated is signed on.

REMOVE stops MTM from accepting commands at a specific terminal or reduces the number of allowable concurrent batch jobs.

device is the terminal to be removed. If the terminal is currently in use, the terminal user is allowed to continue until a SIGNOFF command is entered. When the terminal is removed, MTM closes the logical unit (lu) assignment for the terminal.

BATCH n is the number of concurrent batch streams to be removed. Currently active jobs are allowed to complete execution. If n is greater than the number of current batch streams, the number of concurrent batch streams supported by MTM is reduced to zero. If the new number of batch streams is less than the number of currently active batch jobs, the extra batch jobs are allowed to go to completion.

QUIESCE terminates MTM; new signons are prevented and batch job initiation is stopped. When all terminal users sign off the system and all currently active batch jobs terminate, MTM goes to end of task. Before MTM goes to end of task, information on the availability and usage of and the number of times users had to wait for system resources is output. If MTM discovers that command buffers are missing (the buffer chain has been broken), it pauses to allow user action (such as a panic dump). When the CONTINUE command is entered, MTM goes to end of task code 210.

PURGE removes a batch job from the system and from the batch queue. If the batch job is currently active, it is cancelled.

fd is the file descriptor of the file containing the batch job.

INQUIRE displays the status of a batch job.

fd is the file descriptor of the file containing the batch job. If fd is omitted, the status of all submitted jobs is displayed.

SUBMIT adds a job to the batch queue.

fd is the file descriptor of the file containing the batch job to be added to the batch queue.

ATF changes the accounting transaction file. The current ATF is closed and a new ATF is allocated (if necessary) and assigned.

fd is the file descriptor of the new ATF.

Functional Details:

.MTM STATUS, WAIT, .MTM QUIESCE and .MTM STATUS, USED can be used to determine if the numbers specified at MTM sysgen for command buffers (NCSBFS), console buffers (NCONBFS), sector buffers (NSECTS) and logical units (LINKOPT) are optimal. Increasing the available resources at MTM sysgen can help to eliminate excessive wait conditions. Decreasing resources can eliminate unnecessary memory overhead.

Examples:

Sample output from the .MTM STATUS command:

```
.MTM STA
.MTM: TERMINAL USERS --
.MTM: OS832-BATCH>MTM-MODE   OS3220-BATCH>MTM-MODE   LEE-CT42:>MTM-MODE
.MTM: ROD-CT4C:>MTM-MODE    RICH-CT7A:@MTM-MODE    LAURIE-CT7C:@MTM-MODE
.MTM: JOB M300:OS832.JOB/96 EXECUTING      USER ID=OS832
.MTM: JOB M300:OS3220.JOB/96 EXECUTING      USER ID=OS3220
.MTM: WAITING JOBS = 1
.MTM: 2 BATCH STREAMS
```

Sample output from the .MTM STATUS,WAIT command:

```
.MTM STA,W
.MTM:  TOTAL      PEAK      CURRENT      WAITS
.MTM:    42       32        29           0 CMD BUFFERS
.MTM:    0         0         0           0 CON BUFFERS
.MTM:    7         4         0           0 SEC BUFFEFS
.MTM:   50       10         6           0 LU
```

Where:

TOTAL indicates the maximum number available.

PEAK indicates the maximum number concurrently used.

CURRENT indicates the number in use at the time the command was executed.

WAITS indicates the number of times users were placed in a wait state.

Sample output from the .MTM STATUS,TERMINAL command:

```
.MTM STA,T
.MTM:TERMINAL USERS -
.MTM:OS832-BATCH>MTM-MODE OS3220-BATCH>MTM-MODE LEE-CT42:>MTM-MODE
.MTM:ROD-CT4C:>MTM-MODE RICH-CT7A:@MTM-MODE LAURIE-CT7C:@MTM-MODE
```

Sample output from the .MTM STATUS,BATCH command:

```
.MTM STA,B
.MTM:JOB M300:OS832.JOB/96 EXECUTING USER ID=OS832
.MTM:JOB M300:OS3220.JOB/96 EXECUTING USER ID=OS3220
.MTM:WAITING JOBS = 1
.MTM: 2 BATCH STREAMS
```

This .MTM STATUS,USED command displays, in histogram format, the number of times the indicated number of command buffers were used concurrently. Figure 5-1 indicates, for example, that four command buffers were concurrently used three times, 12 were concurrently used 90 times and 24 were concurrently used 900 times.

USED CMDDB'S	10**1	10**2	10**3	10**4	10**5	10**6	10**7	10**8	10**9
1	*								
2	*								
3	*								
4	***								
5	*****								
6	*****								
7	*****								
8	*****								
9	*****								
10	*****								
11	*****								
12	*****								
13	*****								
14	*****								
15	*****								
16	*****								
17	*****								
18	*****								
19	*****								
20	*****								
21	*****								
22	*****								
23	*****								
24	*****								
25	*****								
26	*****								
27	*****								
28	*****								
29									
30									

Figure 5-1 Sample Output from the .MTM STATUS,USED Command

The .MTM TELL and .MTM MESSAGE commands send the indicated message to the terminal signed on with a userid of SAM.

```
.MTM TEL SAM, YOUR MAG TAPE IS MOUNTED.
```

```
.MTM ME SAM, YOUR MAG TAPE IS MOUNTED.
```

The .MTM BROADCAST command sends the indicated message to all terminals currently assigned to MTM.

```
.MTM BRO MTM IS GOING DOWN IN 15 MINUTES.
```

This .MTM ADD command adds the device CRT2: to MTM.

```
.MTM ADD CRT2:
```

This .MTM ADD command increases the number of batch streams allowed by 2.

```
.MTM ADD BATCH 2
```

This .MTM TERMINATE command terminates the user signed onto CRT8:.

```
.MTM TER CRT8:
```

This .MTM REMOVE command removes CRT10: from MTM upon the next sign off command.

```
.MTM REM CRT10:
```

This .MTM REMOVE command decreases the number of batch streams allowed by 3.

```
.MTM REM BATCH 3
```

The .MTM QUIESCE command terminates MTM after all users have signed off and all batch jobs have terminated.

The following table is output prior to MTM termination.

.MTM QUIESCE

```
.MTM: TOTAL PEAK CURRENT WAITS .MTM: 42 32 29 0 CMD BUFFERS
.MTM: 0 0 0 0 CON BUFFERS .MTM: 7 4 0 0 SEC BUFFERS .MTM: 50
10 6 0 LU .MTM END OF TASK CODE 0
```

The .MTM PURGE command cancels the indicated batch job if active, or removes it from the batch queue if not active.

.MTM PUR MT30:TEST.JOB/8

The .MTM INQUIRE command displays the status of all submitted jobs.

```
.MTM INQ
.MTM:JOB M300:OS832.JOB/96 EXECUTING
.MTM:JOB M300:OS3220.JOB/96 EXECUTING
.MTM:JOB M300:OS3230.JOB/96 WAITING
```

This .MTM INQUIRE command displays the status of the specified job.

.MTM INQ FRED:TASK.JOB/18 JOB FRED:TASK.JOB/18 NOT FOUND

The .MTM SUBMIT command adds the specified job to the batch queue.

.MTM SUB MT30:TEST.JOB/8

5.5 MULTI-TERMINAL MONITOR (MTM) END OF TASK CODES

With the exception of end of task codes 0 and 255, MTM goes to end of task due to a situation that it cannot resolve. The format of the end of task message on the system console is:

.MTM END OF TASK CODE nnn

The MTM end of task codes and meanings are listed in Table 5-1.

TABLE 5-1 MTM END OF TASK CODES

CODE	MEANING
0	Indicates normal end of task.
1	Indicates no message buffers.
2	Illegal entry was removed from the task queue.
3	The task has just been loaded but its task control block (TCB) cannot be located.
4	An SVC6 error has occurred, but the error is not listed in the error table.
6	.MTM START parameter(s) is wrong.
7	A command buffer length of less than 72 or NUSERS less than 1 was specified.
8	The address for the device mnemonic table (DMT) in the volume mnemonic table (VMT) is 0 for a direct access volume.
9	A read or write request was made to record 0 in the batch queue; this request is illegal.
11	The number of I/O classes in the operating system is larger than those in the MTM sysgen.
12	MTM could not assign the system console.
200	The priority scheduling mechanism (PSM) was specified without accounting support.
210	Command buffers are missing upon .MTM QUIESCE (the buffer chain is broken).
254	A user attempted to restart MTM; MTM is not a restartable task.
255	MTM has been cancelled.

5.6 MULTI-TERMINAL MONITOR (MTM) ERROR MESSAGES

The following error messages are generated on MTM.

BATCH ASGN-ERR TYPE=ASGN JOB=fd/act

indicates the lu was not assigned when trying to assign a file at the start of a batch job.

BATCH ASGN-ERR TYPE=BUFF JOB=fd/act

indicates system space was insufficient to obtain a file control block (FCB) when trying to assign a file at the start of a batch job.

BATCH ASGN-ERR TYPE=FUNC JOB=fd/act

indicates an invalid function was detected for a specified device/file when trying to assign it at the start of a batch job.

BATCH ASGN-ERR TYPE=I/O JOB=fd/act

indicates an I/O error was detected on a specified file when trying to assign the file at the start of a batch job.

BATCH ASGN-ERR TYPE=lu JOB=fd/act

indicates an illegal or unassigned lu was detected when trying to assign a file at the start of a batch job.

BATCH ASGN-ERR TYPE=NAME JOB=fd/act

indicates a mismatched filename.ext was detected when trying to assign a file at the start of a batch job.

BATCH ASGN-ERR TYPE=PRIV JOB=fd/act

indicates a mismatched access privilege was detected when trying to assign a file at the start of a batch job.

BATCH ASGN-ERR TYPE=SIZE JOB=fd/act

indicates erroneous record length/size was detected when trying to assign a file at the start of a batch job.

BATCH ASGN-ERR TYPE=VOL JOB=fd/act

indicates no such volume/device was detected when trying to assign a file at the start of a batch job.

BATCH NOT ACTIVE

indicates MTM was started without a batch queue, and a batch request had been made.

BATCH QUEUE ASGN-ERR

indicates the batch queue specified in the START command could not be assigned.

BATCH QUEUE I/O-ERR TYPE=DU- - -ALL BATCH REMOVED- - -

indicates device unavailable was detected on batch queue.

BATCH QUEUE I/O-ERR TYPE=EOF- - -ALL BATCH REMOVED- - -

indicates an end of file (EOF) was detected on batch queue.

BATCH QUEUE I/O-ERR TYPE=EOM- - -ALL BATCH REMOVED- - -

indicates an end of medium (EOM) error was detected on batch queue.

BATCH QUEUE I/O-ERR TYPE=PRTY- - -ALL BATCH REMOVED- - -

indicates parity or a recoverable error was detected on batch queue.

BATCH QUEUE I/O-ERR TYPE=UNRV- - -ALL BATCH REMOVED- - -

indicates an irrecoverable error was detected on batch queue.

BATCH QUEUE LRECL-ERR

indicates a batch queue given in the START command has an invalid record length of less than 40 bytes.

CANNOT ASSIGN AUF FILE

indicates that MTM cannot assign an AUF.

xxxx CMD BUFFERS MISSING

indicates that the number of command buffers specified by
xxxx has been discovered to be missing upon .MTM QUIESCE.

COMMAND BUFFER LENGTH LESS THAN 72

indicates that a command buffer length of less than 72
bytes or NUSERS less than 1 was specified.

dev-I/O ERROR status USER TERMINATED

indicates an irrecoverable error has occurred on a
device; the error status returned is indicated as the
status. MTM automatically signs off a user signed on to
the device.

EXTRANEIOUS OPERANDS

indicates an MTM command has too many operands.

FILE DESCRIPTOR ERROR

indicates the fd specified in the command is omitted or
its format is incorrect.

INVALID COMMAND

indicates the MTM command entered is unrecognized.

INVALID PARAMETER

indicates a command has been entered with an invalid
parameter or bad syntax or the device cannot be found.

voln:MTMASST.TSK/S HAS WRONG REVISION LEVEL

indicates that the revision level for .MTMASST is
incorrect. Execution of the .MTMASST task is cancelled.

voln:MTMASST.TSK/S NOT FOUND OR LOAD ERROR

indicates that the .MTMASST task could not be executed
because the MTMASST.TSK/S file could not be found, or an
error occurred when the task was loaded.

NO BUFFER AVAIL FOR CMD

indicates that all command and message buffers are full.

START OPTIONS FORM-ERR

indicates an invalid format in the START command.

START OPTIONS PARM-ERR

indicates an invalid START parameter.

TERMINAL ASSIGN ERROR

indicates an attempt to add a new terminal to the system has failed. The device is not recognized or not available.

TERMINALS OR BATCH EXCEED MAXIMUM

indicates the number of concurrent batch streams allowed exceeds 64, the number of terminals exceeds 64, or the number of batch streams and terminals exceed the NUSERS established at MTM sysgen time.

USERID NOT FOUND

indicates a current user has not been found with this identification.

CHAPTER 6
MULTI-TERMINAL MONITOR (MTM) OPERATION IN A
MODEL 3200MPS SYSTEM MULTIPROCESSING ENVIRONMENT

6.1 INTRODUCTION

When an MTM operates in a Perkin-Elmer Model 3200MPS System multiprocessing environment, an MTM subtask can be executed on any processor in the system. The terminal user can manually assign a task to a specific processor queue by:

- entering an OPTION LPU command after loading the task (see the OS/32 Multi-Terminal Monitor (MTM) Reference Manual), or
- linking the task with the LPU= option (see the OS/32 Link Reference Manual).

Whether or not the above methods enhance system performance depends upon the task load of the particular processor assigned to the task in relation to the task loads of other processors in the system. Since a single terminal user may not be familiar with the requirements of the entire MTM installation, he may be unable to choose the processor that will result in optimum system performance. An improper auxiliary processing unit (APU) task assignment may even inhibit system performance.

To overcome the disadvantages of manual APU task queue assignment, Perkin-Elmer provides load leveling executive (LLE) software. This software supplies the MTM system administrator with an automatic method of APU task queue assignment that is designed to take full advantage of the multiprocessing capabilities of the Model 3200MPS System. LLE automatically directs MTM subtasks either to the central processing unit (CPU) ready queue or to APU queue 0 depending on the run-time behavior of these subtasks. The subtasks scheduling priority is calculated by the priority scheduling mechanism (PSM) embedded in MTM.

Once activated, LLE forces all subtasks without the supervisor call 6 (SVC6) control privilege (see Section 3.4.9) to use logical processing unit (LPU) 0 and operates only on these subtasks. All LLE operations on the subtasks are transparent to the terminal user.

LLE also collects system related information and prepares statistics displayed with the .MTM LLE DISPLAY command.

This chapter describes the operation of the LLE software and the system administrator's responsibilities for initializing, controlling and optimizing LLE for optimum system performance.

6.2 OPERATION OF THE LOAD LEVELING EXECUTIVE (LLE)

MTM subtasks running in a Model 3200MPS System can be made subject to execution on the CPU or an APU. Tasks that are subject to execution on the CPU are organized into a special task scheduling queue called the CPU ready queue. Tasks subject to execution on an APU are organized in one of the APU queues. For load leveling purposes, a CPU/APU shared queue called queue 0 is used. Queue 0 is serviced by all APUs assigned to queue 0 and the CPU; the CPU, however, will only service this queue when its own ready queue is empty.

The task requests for operating system services (SVCs, fault and trap processing) are fulfilled on the CPU ready queue. Whenever a service is required by a task assigned to queue 0, the task is transferred (passed back) to the CPU ready queue. Upon service completion, the task is passed back to queue 0 by the operating system. The LLE minimizes the traffic between the two queues (APU/CPU thrashing) according to the tasks' run-time behavior. Essentially, the computation-intensive tasks are directed to LPU 0, which is always mapped to APU queue 0. Tasks can now be executed by any processor in the system (thus achieving load leveling). The service-intensive tasks (e.g., I/O-intensive) are directed to the CPU ready queue.

LLE executes as a subtask of MTM and has a taskid of .MTMLLE. MTM loads, starts and controls the LLE task via commands entered at the system console. The task priority of .MTMLLE is 10. Once LLE is started, it runs periodically at a PSM interval controlled by the system operator.

6.2.1 Queue 0 Eligibility

LLE employs a queue 0 eligibility algorithm to determine which MTM subtasks are to be directed to queue 0. Queue 0 eligibility of a task, E, is computed as follows:

$$E = 100\% * (P - NP * TP) / PM$$

Where:

P	is the task processor time (CPU and APU user time) during the last PSM interval.
NP	is the number of task transfers (passbacks) from APU to CPU during the last interval.
TP	is passback overhead time (i.e., an estimate of time spent by OS/32 to handle passback).
PM	is the maximum task processor time allowed in a PSM interval.

If the above equation for E yields a negative value, E is set to 0. For active tasks that did not obtain any processor time in the last PSM interval, the value of E calculated at the previous PSM interval is used.

Hence, a task that consumed all of the processor time allowed in a PSM interval and had no passbacks during the last PSM interval is 100% queue 0 eligible for the next PSM interval. During a PSM interval, a task that uses 100ms of CPU and APU time and makes 15 passbacks with an estimated 1ms of overhead each, will be 85% queue 0 eligible.

OS/32, the APUs and PSM supply the run-time behavior information needed by LLE to determine queue 0 eligibility. The resultant queue 0 eligibility for the task is compared against a fixed eligibility level established by the system operator when LLE is started. If the task's queue 0 eligibility is greater than the fixed level, the task is directed to queue 0. If it is equal to or less than the fixed level, the task is directed to the CPU ready queue.

6.2.2 Prevention of Central Processing Unit (CPU) Overload

The queue 0 eligibility algorithm checks whether the CPU was overloaded, (i.e., did not service queue 0 during the last PSM interval). This is accomplished by using the CPU queue 0 access count, which is incremented by the operating system. If the CPU was overloaded, one task with queue eligibility less than the fixed level is directed to queue 0 for the next interval.

6.2.3 Error Termination

During execution, the LLE task continually checks queue 0 to ensure that at least one APU is assigned to the queue. If no APU is assigned to queue 0, LLE goes to end of task after sending the following message to the system console:

NO APU ASSIGNED TO QUEUE-LLE TERMINATES

If execution of the LLE task terminates or is suspended because of an error, MTM immediately discontinues communication with the task and informs the operator of LLE error status by sending a message to the system console. MTM then redirects all subtasks on queue 0 to the CPU ready queue. Until LLE is reactivated via an LLE control command, all MTM subtasks are started from the CPU ready queue. This process is entirely transparent to the MTM terminal user.

6.3 OPTIMIZATION AND CONTROL OF LOAD-LEVELING

The system operator is able to optimize load-leveling for a particular installation by controlling the following values:

- PSM interval length
- Maximum number of I/O operations allowed for a task during the PSM interval
- Maximum processor time allowed for a task during the PSM interval
- Queue 0 eligibility limit

These values are displayed and changed using the .MTM PSM and .MTM LLE commands (see Section 6.4). The decisions to change the above values are based on:

- Establishing criteria for load-leveling at the given installation; e.g., maximum system throughput with a limit on user response time, maximum speed of response with a limit on how low the system throughput can fall or minimum system overhead, etc.

- Sampling applicable accounting data to determine a favorable change of parameters upon varying the above values; e.g., increasing PSM interval length generally reduces overhead, but increases the response delay; increasing the next two values increases throughput for the I/O- or computation-intensive tasks, respectively, but will decrease response time; increase of the eligibility limit lowers the passback overhead, but also decreases APU utilization, etc.

The system operator is able to monitor and control LLE execution and initial task direction to the appropriate queue. This way the traffic optimization between the queues provided by the LLE is enabled or disabled and the initial task distribution is set. These functions are performed using the .MTM LLE command.

6.4 LOAD-LEVELING OPTIMIZING AND CONTROL (.MTM LLE) COMMANDS

The commands used by the system administrator to optimize and control operation of the LLE are:

- .MTM LLE CPU
- .MTM LLE DISPLAY
- .MTM LLE ELIGIBLE
- .MTM LLE LPU
- .MTM LLE OFF
- .MTM LLE ON

The commands used by the system administrator to control the PSM are:

- .MTM PSM
- .MTM PSM INTERVAL
- .MTM PSM I/O
- .MTM PSM MAXPROCESSOR

.MTM LLE

6.4.1 .MTM LLE Command

The .MTM LLE command enables the system operator to control the LLE.

Format:

.MTM LLE { LPU
CPU
DISPLAY
OFF
ON
ELIGIBILITY [(n,F)
(D)
(,N)] }

Parameters:

- LPU** When LLE is active, this command will cause MTM to place all of its subtasks (which are able to run on an APU) on queue 0 (LPU 0 directed). When LLE is not active, this command will be rejected and the MTM subtasks will be placed on the CPU ready queue.
- CPU** When LLE is active, this command will cause MTM to place all of its subtasks on the CPU ready queue.
- DISPLAY** This command displays current variable/fixed threshold values, PSM values, CPU and APU loads and queue information to the system console.
- OFF** This command removes the LLE from the system; all future starts of MTM subtasks will be on the CPU ready queue.
- ON** This command will load and start the LLE (taskid = .MTMLLE). Eligibility is set to its default which is 1%.

ELIGIBILITY This command controls the eligibility value and/or the evaluation logic.

n is the fixed queue 0 eligibility value in percent. The range of n is between 1 and 99.

F is the fixed queue 0 eligibility mode for the LLE.

D is the dynamic queue 0 eligibility mode for the LLE.

N is the nooperation mode for the LLE.

Functional Details:

The .MTM LLE commands require that MTM is sysgened with the PSM (see Section 2.4.21).

WARNING

THE LLE DOES NOT CONTROL ALL OF THE TASKS CAPABLE TO EXECUTE ON APU QUEUE 0. NAMELY, IT DOES NOT CONTROL MTM SUBTASKS WITH THE SVC6 COMMUNICATIONS PRIVILEGE LOADED FROM THE OPERATOR CONSOLE. THIS MAY LEAD TO MONOPOLIZATION OF APUs ASSIGNED TO QUEUE 0 BY SUCH UNCONTROLLABLE TASKS AND THEREBY DEGRADES PERFORMANCE OF MTM SUBTASKS.

After the .MTM LLE OFF command is entered, MTM subtasks which are on queue 0 are transferred back to the CPU ready queue.

After the .MTM LLE ON command is entered, MTM checks that the PSM was enabled at MTM sysgen. If the PSM was not enabled, this command is rejected. If the PSM was enabled, MTM issues an SVC13 to set queue 0 to enforced priority order and loads the LLE. MTM then checks the LLE revision level to ensure that the proper terminal user (TUB) structure is used. An error message is then output to the system console to inform the user that LLE has been cancelled.

The .MTM LLE ELIGIBILITY command, when set to the fixed mode, enables the system operator to establish a fixed queue 0 eligibility level. The range of n is between 1 and 99, which indicates the fixed queue 0 eligibility value. This value sets the fixed queue 0 eligibility value for the LLE and is used to determine if a task is to be placed on either queue 0 or the CPU ready queue.

| When the .MTM LLE ELIGIBILITY command is set to the dynamic mode,
| LLE switches to a CPU-overload algorithm to determine queue 0
| eligibility in cases of CPU overload. In cases where the CPU is
| underloaded or up to 100% loaded, no dynamic threshold value
| change is necessary. The LLE maintains enough queue 0
| eligibility tasks on queue 0 to keep APUs and the CPU busy.

| CPU overload is a critical case for the LLE. The LLE is informed
| of a CPU overload condition when the counter for CPU access to
| queue 0 is zero for the last PSM interval. The LLE reacts with
| a change of the variable threshold for queue 0 eligible tasks to
| force more tasks less queue 0 eligibility to be moved to queue 0.
| The formula is as follows:

$$\text{CPU unload} = [(\text{CPU+APU time}) - \text{passbacks}] * \text{overhead}$$

| Where:

| CPU+APU time is the total processor time used by the
| subtask during the last PSM interval.

| passbacks is the number of passbacks that occurred in
| the last PSM interval for the subtask.

| overhead is the CPU overhead time required to complete
| one passback for the subtask.

| If the CPU unload value for the task is greater than zero, the
| CPU will be unloaded by this value if the task executes on an
| APU. The LLE never moves a task with a negative or zero value to
| queue 0 because this would cause the CPU to be even more
| overloaded.

| When the .MTM LLE ELIGIBILITY command is set to the nooperation
| mode, the LLE will not cancel the task but will store the task in
| memory for future use. Otherwise the eligibility n value is lost
| and set to its default when LLE is turned back on.

| Examples:

| This example causes MTM subtasks under LLE control to be placed
| on queue 0:

| .MTM LLE LPU

The .MTM LLE DISPLAY command outputs the following information to the system console:

```
14:56:45 .MTM:PSM: INTERVAL= 4 I/O= 335 MAXPROCESSOR=25 %
14:56:45 .MTM:
14:56:45 .MTM:LLE MODE=DYNAMIC ELIGIBLE= 1.0 %
14:56:45 or .MTM:LLE MODE=FIXED ELIGIBLE= 0.0 %
14:56:45 or .MTM:LLE MODE=NOOPERATION
14:56:45 .MTM:
14:56:45 .MTM:LOAD: LAST 4 SEC LAST 40 SEC SINCE START
14:56:45 .MTM: CPU 18.8 % 4.5 % 6.8 %
14:56:45 .MTM: APUL 0.0 % 0.0 % 0.0 %
14:56:45 .MTM:
14:56:45 .MTM:MTM: SUBTASK INFORMATION
14:56:45 .MTM:
14:56:45 .MTM: LPU 0 DIRECTED 0 TASKS
14:56:45 .MTM: CPU DIRECTED 4 TASKS
14:56:45 .MTM: CPU READY QUEUE 0 TASKS
14:56:45 .MTM: CPU REC. QUEUE 0 TASKS
14:56:45 .MTM: QUEUE 0 4 TASKS
14:56:45 .MTM: EXEC/WAIT/ROLLED 0 TASKS
```

This example removes LLE from the system:

```
.MTM LLE OFF
```

The following example loads the MTMLLE.TSK/S from the default system volume:

```
.MTM LLE ON
```

Table 6-1 lists the error messages output by these commands.

TABLE 6-1 LLE ERROR MESSAGES

ERROR MESSAGE	AFFECTED COMMANDS	MEANING
ERROR: NO PSM OR NON MPS 3200 SYSTEM	.MTM LLE CPU .MTM LLE DISPLAY .MTM LLE ELIGIBLE .MTM LLE LPU .MTM LLE OFF .MTM LLE ON	The entered command is rejected because PSM was not sysgered or the command was entered on a system other than a Model 3200MPS.
WARNING: LLE STATUS=OFF	.MTM LLE CPU .MTM LLE ELIGIBLE .MTM LLE LPU .MTM LLE DISPLAY .MTM LLE OFF .MTM LLE ON	The command was entered when the LLE task was in an inactive state. The command will have no effect on task direction to queues until the LLE task is activated. The command is rejected because the LLE task is in the inactive state.
WARNING: LLE STATUS=UNDEFINED	.MTM LLE CPU .MTM LLE ELIGIBLE .MTM LLE LPU .MTM LLE DISPLAY .MTM LLE OFF .MTM LLE ON	The command was entered when the LLE task was in an undefined state. The command will have no effect on task direction to queues until the LLE task is activated. The command is rejected because the LLE task is in an undefined state.
WARNING: LLE STATUS=PAUSED	.MTM LLE CPU .MTM LLE ELIGIBLE .MTM LLE LPU .MTM LLE DISPLAY .MTM LLE OFF .MTM LLE ON	The command was entered when the LLE task was in a paused state. The command will have no effect on task direction to queues until the LLE task is reactivated. The command is rejected because the LLE task is in the paused state.
WARNING: LLE STATUS=TIMEOUT (TASK QUEUE OVERFLOW)	.MTM LLE CPU .MTM LLE ELIGIBLE .MTM LLE LPU .MTM LLE DISPLAY .MTM LLE OFF .MTM LLE ON	The command was entered when the LLE task was in a time-out state due to overflow of .MTM LLE task event queue. The command will have no effect on task direction to queues until the LLE task is reactivated. The command is rejected because the LLE task is in the time-out state.
WARNING: LLE STATUS=MEMORY FAULT (instruction address/memory address)	.MTM LLE CPU .MTM LLE ELIGIBLE .MTM LLE LPU .MTM LLE DISPLAY .MTM LLE OFF .MTM LLE ON	The command was entered when the LLE task was in a suspended state due to a memory fault. The command will have no effect on task direction to queues until the LLE task is reactivated. The command is rejected because the LLE task is in a suspended state due to a memory fault.

TABLE 6-1 LLE ERROR MESSAGES (Continued)

ERROR MESSAGE	AFFECTED COMMANDS	MEANING
WARNING: LLE STATUS=ILLEGAL INSTRUCTION (address)	.MTM LLE CPU .MTM LLE ELIGIBLE .MTM LLE LPU .MTM LLE DISPLAY .MTM LLE OFF .MTM LLE ON	The command was entered when the LLE task was in a suspended state due to an illegal instruction fault. The command will have no effect on task direction to queues until the LLE task is reactivated. The command is rejected because the LLE task is in a suspended state due to an illegal instruction fault.
ERROR: FILE DESCRIPTOR ERROR	.MTM LLE ON	The command is rejected because .MTM cannot load the LLE task due to an fd error.
ERROR: LLE STATUS=ON	.MTM LLE ON	The command is ignored because an attempt was made to start the LLE task when the task is already active.
ERROR: LLE STATUS=yyyy AT erradd/LLE.SAD AT segadd	.MTM LLE CPU .MTM LLE DISPLAY .MTM LLE ELIGIBILITY .MTM LLE LPU .MTM LLE OFF .MTM LLE ON	The command cannot be performed; the LLE may be in one of the following error states where yyyy is replaced by: ???? = undefined (paused, looping, etc.) NAPU = no APUs assigned to queue 0 ILGF = internal logic fault ARFT = arithmetic fault PWFT = power fail trap DFFT = data format fault IINS = illegal instruction MMAL = machine malfunction erradd = the absolute address where the error occurred. segadd = the absolute address where .MTM LLE UDL starts. To recover from those errors, use the .MTM LLE OFF command, then the .MTM LLE ON command.
ERROR: LOAD OR START ERROR, SVC6, STATUS= X'yyyy'	.MTM LLE ON	Indicates that the load of the LLE returned an error status for the SVC6. yyyy is the hex status returned by the SVC6 handler.
ERROR: voln: MTMLLE.TSK/S ASSIGN ERROR	.MTM LLE ON	Indicates that the MTMLLE.TSK/S on the default system volume (voln:) could not be assigned to .MTM.
ERROR: voln: MTMLLE.TSK/S HAS WRONG REVISION LEVEL	.MTM LLE ON	Indicates that the MTMLLE.TSK.S on the default system volume (voln:) has a different revision than the .MTM task currently executing.

.MTM PSM

6.4.2 .MTM PSM Command

This command enables the system operator to control the PSM.

Format:

.MTM PSM {
INTERNAL n₁
I/Q n₂
MAXPROCESSOR n₃}

Parameters:

PSM indicates that this command is directed to the PSM handler within .MTM. If no further parameter is specified, the current PSM values are displayed on the system console.

INTERVAL selects the interval value for change.

n₁ is the time interval in seconds between the PSM priority recomputing cycles. The range for n is 1 through 10.

I/O selects the I/O count for change.

n₂ sets the maximum number of I/O (performed in the last PSM interval) that are used to recompute the new priority for an MTM subtask. The range for n is 1 through 9,999.

MAXPROCESSOR selects the MAXPROCESSOR value for change.

n₃ is used as a percent value in PSM to determine the MTM subtask priority. If an MTM subtask uses more than this value of processor time (in the last PSM interval), it's priority is reset to the lowest PSM priority.

Functional Details:

The new PSM values are displayed after each command to the system console. The defaults set up at the start of MTM are as follows:

Interval n = 4 sec / I/O n = 500 / MAXPROCESSOR n = 25 %

Example:

This example selects an I/O count of 335 and the following information is returned:

```
.MTM PSM I/O 335
10:02:20 .MTM:CURRENT PSM: INTERVAL=04 SEC I/O=0335 MAXPROCESSOR=25 %
```

Table 6-2 lists the error messages output by these commands.

TABLE 6-2 PSM ERROR MESSAGES

ERROR MESSAGE	AFFECTED COMMANDS	MEANING
INVALID PARAMETER	.MTM PSM INTERVAL .MTM PSM I/O .MTM PSM MAXPROCESSOR	The given parameter n is invalid. Check the value specified and correct accordingly.
PSM NOT SYSGENED	.MTM PSM .MTM PSM INTERVAL .MTM PSM I/O .MTM PSM MAXPROCESSOR	The entered command is rejected because the PSM was not system generated (sysgened).
MISSING OPERAND	.MTM PSM INTERVAL .MTM PSM I/O .MTM PSM MAXPROCESSOR	The entered command has a missing operand. Reenter the command specifying a value for n.
EXTRANEIOUS OPERAND	.MTM PSM .MTM PSM INTERVAL .MTM PSM I/O .MTM PSM MAXPROCESSOR	The entered command has an extraneous operand. Reenter the command omitting the extraneous operand.

6.5 LOAD-LEVELING EXECUTIVE (LLE) OPERATING TIPS

Before turning LLE on, the operator should ensure that PSM has been sysgened into the system. See Chapter 2 for information on the MTM sysgen option. LLE overrides LPU task assignment supplied via MTM or Link; therefore, the operator should broadcast a message to all MTM users that automatic load leveling is in effect. The user must have APU and SVC6 control privileges to be able to access LPU numbers other than 0. If the MTM terminal user wishes to run a task on an APU that is not assigned to queue 0, that task can be run from the system console.

APPENDIX A
MULTI-TERMINAL MONITOR (MTM)
SYSTEM GENERATION (SYSGEN) OPTIONS

OPTIONS:

CDBUFSIZ EQU n	Size of sector buffers
CONBFSZ EQU n	Size of console buffers
CSBFSZ EQU n	Size of command buffers
NCONBFS EQU n	Number of console buffers
NCSBFS EQU n	Number of extra command buffers
NCSSLVS EQU n	Number of command substitution system (CSS) levels
NMSGs EQU n	Number of message buffers
NSECTS EQU n	Number of sector buffers
NUSERS EQU n	Number of concurrent users
PRCLIMIT EQU n	Processor time limit
QUESIZE EQU n	Size of task queue
SGN.ACT EQU n	Accounting support/number of classes
SGN.BTCP EQU $\left. \begin{matrix} 0 \\ 1 \end{matrix} \right\}$	Batch job pause

SGN.MVOL EQU C'name'	Default user volume
SGN.PRIO EQU n	Batch priority
SGN.SCHR EQU n	Subtask scheduling
SGN.ROLL EQU $\left. \begin{matrix} 0 \\ 1 \end{matrix} \right\}$	Tasks rollable
SGN.SPL EQU n	SPL/32 support/number of buffers
SGN.VAR EQU n	Numbers of variables
SGN.@GLO EQU n	Number of new global variables
SGN.@INT EQU n	Number of internal variables
SGN.\$HSP EQU $\left. \begin{matrix} 0 \\ 1 \end{matrix} \right\}$	HASP interface

MACROS:

IOTAB - I/O class number, identification, limit:

[label] IOTAB (classno,classid [iocount])₀
('...',classno,classid [iocount])_n

PRMTDEV - device to which prompt sequence applies:

[label] PRMTDEV (pname₁,dev₁ [, ..., dev_n])₁
[, ..., (pname_n,dev₁ [, ..., dev_n])_n]

[label] PRMTDEV (SIGMA,dev₁ [, ..., dev_n])

PRMTSEQ - prompt sequence; command, task, build:

[label] PRMTSEQ (SIGMA)

[label] PRMTSEQ (pname₁, (com-req, x), (task-exec, x),
(task-read, x), (build-read, x))₁
[, ...,
(pname_n, (com-req, x), (task-exec, x),
(task-read, x), (build-read, x))_n]

[label] PRMTSEQ (STANDARD, (com-req, x), (task-exec, x),
(task-read, x), (build-read, x))₁

UIOLIM - I/O limits, all I/O classes per user class:

[label] UIOLIM (u_{0io0t}, ..., u_{0iont}), (u_{0io0b}, ..., u_{0ionb})
(u_{niont}, ..., u_{niont}), (u_{nio0b}, ..., u_{nionb})

UPRCLIM - processor time limits for I/O classes: |

[label] UPRCLIM limit_{0t}, limit_{0b}, ..., ..., limit_{nt}, limit_{nb} |

VERSION - MTM parameter file identification:

[label] VERSION 'id'

APPENDIX C
AUTHORIZED USER UTILITY MESSAGE SUMMARY

taskid > for CON:
or
---> under MTM

is the prompt, generated for an interactive device, indicating an input request.

ACTUTILITY rr-uu

identifies the Account Utility. The revision level is rr; the update level is uu. This message is output only at the initial start of the Account Utility.

ACCOUNT UTILITY RUN xx/xx/xx yy:yy:yy

appears on the system console when the Account Utility is started. It informs the operator that the Account Utility has been run.

***SET DATE & TIME

is the date request that pauses the Account Utility so the current date and time can be set for the system. The system CONTINUE command transfers control back to the Account Utility. This message is output when the date and time are not set.

***ACCOUNT NUMBER xxx WAS INACTIVE

indicates the requested command could not be performed because the indicated account record was inactive. The indicated account number is xxx.

*****ACCOUNT NUMBER xxx WAS ACTIVE**

indicates the requested command could not be performed because the indicated account record was already active. The indicated account number is xxx.

ACCOUNTS=n

indicates the maximum account number on the current AUF.

*****xxxxxx IS THE CURRENT MODE**

indicates the output from the MODE command. The name of the current mode (CREATE, UPDATE or REPORT) is xxxxxx.

?

is the question mark, along with some of the following messages, used to denote which parameter in the command caused the error. The command and a pointer (?) to the location of the error is displayed to the command input device.

*****ILLEGAL INPUT COMMAND**

indicates the command word or abbreviation could not be found in the table of legal commands.

*****WRONG MODE FOR INPUT COMMAND**

indicates the current mode does not allow the input command.

*****ILLEGAL COMMAND FORM**

indicates the input command uses too many separators or a separator other than a comma or equal character.

*****ILLEGAL PARAMETER**

indicates the input command is missing a required parameter or has too many parameters.

*****ILLEGAL START PARAMETER**

indicates a START command word or abbreviation was not found in the START command table.

APPENDIX C
AUTHORIZED USER UTILITY MESSAGE SUMMARY

taskid > for CON:
or
----> under MTM

is the prompt, generated for an interactive device,
indicating an input request.

ACTUTILITY rr-uu

identifies the Account Utility. The revision level is rr;
the update level is uu. This message is output only at the
initial start of the Account Utility.

ACCOUNT UTILITY RUN xx/xx/xx yy:yy:yy

appears on the system console when the Account Utility is
started. It informs the operator that the Account Utility
has been run.

***SET DATE & TIME

is the date request that pauses the Account Utility so the
current date and time can be set for the system. The system
CONTINUE command transfers control back to the Account
Utility. This message is output when the date and time are
not set.

***ACCOUNT NUMBER xxx WAS INACTIVE

indicates the requested command could not be performed
because the indicated account record was inactive. The
indicated account number is xxx.

***ACCOUNT NUMBER xxx WAS ACTIVE

indicates the requested command could not be performed because the indicated account record was already active. The indicated account number is xxx.

ACCOUNTS=n

indicates the maximum account number on the current AUF.

***xxxxxx IS THE CURRENT MODE

indicates the output from the MODE command. The name of the current mode (CREATE, UPDATE or REPORT) is xxxxxx.

?

is the question mark, along with some of the following messages, used to denote which parameter in the command caused the error. The command and a pointer (?) to the location of the error is displayed to the command input device.

***ILLEGAL INPUT COMMAND

indicates the command word or abbreviation could not be found in the table of legal commands.

***WRONG MODE FOR INPUT COMMAND

indicates the current mode does not allow the input command.

***ILLEGAL COMMAND FORM

indicates the input command uses too many separators or a separator other than a comma or equal character.

***ILLEGAL PARAMETER

indicates the input command is missing a required parameter or has too many parameters.

***ILLEGAL START PARAMETER

indicates a START command word or abbreviation was not found in the START command table.

***DUPLICATE START PARAMETER

indicates the START command contains a duplicate command word or more than one mode request.

***ILLEGAL FILE DESCRIPTOR

indicates the file descriptor (fd) used is syntactically incorrect.

***MISSING START PARAMETER

indicates at least one START parameter is missing from the specified mode.

***ALLOCATION/ASSIGNMENT ERROR

indicates a file cannot be allocated because of a filename conflict or the filename specified for the assignment could not be found on the specified disk.

***lu xx NOT ASSIGNED

indicates a requested function could not be performed because the logical unit (lu) xx was not assigned. The utility is paused and control transfers to the system. The system CONTINUE command continues the utility after the assignment.

***INPUT DEVICE ERROR xx on LOGICAL UNIT yy

indicates an input request has an input/output (I/O) error on device xx assigned to lu yy. The Authorized User Utility pauses. The system CONTINUE command transfers control back to the utility.

***OUTPUT DEVICE ERROR xx ON LOGICAL UNIT yy

indicates an output request has the I/O error on device xx assigned to lu yy. The Authorized User Utility pauses. The system CONTINUE command transfers control back to the utility.

***MISSING PARAMETER

indicates the input command is missing a required parameter. On input errors, where the form of the command or its characters are incorrect, the command and a pointer (?) to the location of the error is displayed to the user console.

***MODE NOT ALLOWED

indicates an attempt to perform an UPDATE or a REPORT was made from an account other than 255.

***WARNING ACCOUNT 255 DOES NOT EXIST

indicates an attempt to end the Account Utility when account 255 does not exist.

APPENDIX D
ACCOUNTING TRANSACTION FILE (ATF) RECORD FORMATS

The data structure of the ATF is defined by the \$ATF data structure macro contained in SYSSTRUC.MLB. The fields and functional values in the ATF account records have standard symbolic names. These names may be accessed by coding and expanding the \$ATF data structure macro (see the Common Assembly Language Macro/32 Processor (CAL Macro/32) and Macro Library Utility Reference Manual). Each of the six ATF account record types is associated with a particular type of account activity. The record length of an ATF record varies with the number of input/output (I/O) classes supported by the system. The record formats shown in this appendix are 128 bytes and accept entries for the four default I/O classes. The record types are listed in Table D-1.

TABLE D-1 ATF RECORD
TYPES

TYPE	RECORD ACTIVITY
1	Signon
2	Signoff
3	Task completion
4	Task data overflow
5	File deletion
6	File renaming

These six record formats are illustrated in the following figures.

0(0)	Record Type (ATF.TYPE)	2(2)	Account Number (ATF.ACCT)
4(4)	Current date and time (ATF.CRTM)		
8(8)	User identification		
12(C)	(ATF.UID)		
16(10)	Job serial number (ATF.JOB)		
20(14)			
24(18)			
28(1C)	File descriptor (fd) of command input file (ATF.UFD)		
32(20)			
36(24)			
40(28)	Date and time job was submitted to batch queue (ATF.SUBT)		
44(2C)			
48(30)	Filler		
.			
.			
.			
116(74)			
120(78)	Filler		
124(7C)			

Figure D-1 ATF Format for Signon Account Record Type

0(0)	Record type (ATF.TYPE)		2(2)	Account Number (ATF.ACCT)
4(4)	Current date and time (ATF.CRTM)			
8(8)	User identification			
12(C)	(ATF.UID)			
16(10)	Job serial number (ATF.JOB)			
20(14)	Number of tasks run (ATF.TASK)		22(16)	Number of CSS calls (ATF.CSS)
24(18)	Number of commands entered (ATF.CMDS)		26(1A)	Number of display files (ATF.DISF)
28(1C)	Elapsed time (ATF.ELPS)			
32(20)	Filler			
36(24)				
⋮				
116(74)				
120(78)	Filler			
124(7C)				

Figure D-2 ATF Format for Signoff Account Record Type

0(0)	Record type (ATF.TYPE)	2(2)	Account Number (ATF.ACCT)
4(4)	Current date and time (ATF.CRTM)		
8(8)	User identification		
12(C)	(ATF.UID)		
16(10)	Job serial number (ATF.JOB)		
20(14)	Time task was started (ATF.STAR)		
24(18)			
28(1C)			
32(20)	Id of task (ATF.TFD)		
36(24)			
40(28)	Account number (ATF.ACT)	42(2A)	End of task code (ATF.RC)
44(2C)	User-supplied information		
48(30)	(ATF.USER)		
52(34)	Impure segment size (ATF.IMP)	54(36)	Pure segment size (ATF.PUR)
56(38)	System space (ATF.SYS)	58(3A)	Roll count (ATF.ROLL)
60(3C)	User CPU time (ATF.UCPU)		
64(40)	SVC CPU time (ATF.SCPU)		
68(44)	ROLL CPU time (ATF.RCPU)		
72(48)	User APU time (ATF.UAPU)		
76(4C)	Wait time (ATF.WTIM)		
80(50)	Roll queue time (ATF.RLTM)		
84(54)	Input/output (I/O) count for class 0 (ATF.IOCT)		
88(58)	Number of bytes transferred for class 0 (ATF.IOXF)		
92(5C)	I/O Count for class 1		
96(60)	Number of bytes transferred for class 1		
100(64)	I/O Count for class 2		
104(68)	Number of bytes transferred for class 2		
108(6C)	I/O Count for class 3		
112(70)	Number of bytes transferred for class 3		
116(74)			
120(70)	Filler		
124(7C)			

Figure D-3 ATF Format for Task Completion (ATYP.TSK) and Task Data Overflow Account Record Types

0(0)	Record type (ATF.TYPE)		2(2)	Account number (ATF.ACCT)
4(4)	Current date and time (ATF.CRTM)			
8(8)	Volume name (AFT.VOL)			
12(C)	Filename (first Word) (ATF.FNAM)			
16(10)	Message pointer for SVC6 (ATF.MSGP)			
20(14)	Filename (second word) (ATF.FNM2)			
24(18)	Extension (ATF.EXT)			
28(1C)	Number of records (ATF.RECS)			
32(20)	Number of sectors (ATE.SECONDS)			
36(24)	File creation date (ATF.CDAT)			
40(28)	New filename			
44(2C)	(ATF.NWFN)			
48(30)	Extension (ATF.NWEX)			
52(34)	Account number (ATF.NWAC)		54(36)	Filler
56(38)	ID of task			
60(3C)	that renamed file (ATF.TKID)			
64(40)	Filler			
124(7C)				

Figure D-4 ATF Format File Renaming Account Record Type

0(0)	Record type (ATF.TYPE)	12(2)	Account number (ATF.ACCT)
4(4)	Current date and time (ATF.CRTM)		
8(8)	Volume name (ATF.VOL)		
12(C)	Filename (first word) (ATF.FNAM)		
16(10)	Message pointer for SVC6 (ATF.MSGP)		
20(14)	Filename (second word) (ATF.FNM2)		
24(18)	Extension (ATF.EXT)		
28(1C)	Number of records (ATF.RECS)		
32(20)	Number of sectors (ATE.SECS)		
36(24)	File creation date (ATF.CDAT)		
40(28)			
44(2C)			
48(30)	Filler		
52(34)			
56(38)	ID of task that deleted file		
60(3C)	(ATF.TKID)		
64(40)			
	Filler		
124(7C)			

Figure D-5 ATF Format for File Deletion Account Record Type

APPENDIX E
MULTI-TERMINAL MONITOR (MTM) OPERATOR COMMAND SUMMARY

LOAD .MTM, fd

TASK .MTM

START ,AUF=fd [ATF=fd] [,TERMINAL=(fd₁ [,fd₂,...,fd₃])]]
[BATCH=(n, fd)] [,PAGE=fd]

.MTM	STATUS	{	WAIT	}	
		TERMINAL			
		BATCH			
		USED [,fd ₁]			
		current batch and terminal			
	{	TELL	}userid message		
		MESSAGE			
				BROADCAST message	
	ADD	{	device:	}	
			BATCH n [,fd]		
			TERMINATE device:		
REMOVE	{	device:	}		
		BATCH n			
			QUIESCE		
			PURGE fd		
			INQUIRE [fd]		
			SUBMIT fd		
			ATE fd		

.MTM LLE {

- LPU
- CPU
- DISPLAY
- OFF
- ON
- ELIGIBILITY [{ n, F }
D
, N]

 }

.MTM PSM {

- INTERVAL n
- I/Q n
- MAXPROCESSOR n

 }

INDEX

A		B	
Account limit	3-9	BATCH parameter	5-2
Account number	3-3	.MTM ADD	5-5
	4-7	.MTM REMOVE	5-6
	4-11	MTM START command	5-2
Accounting data		Batch processing	
batch job	4-5	PRC limit	2-16
task-related	4-7	BROADCAST parameter	5-5
terminal session	4-5		
terminal user access	4-12		
Accounting facility	4-1	C	
system requirements	4-2	CDBUFSIZ	2-6
Accounting times and formats		CHANGE command	3-12
elapsed time	4-4	Command buffers	2-7
processor time	4-4		2-12
processor time left	4-5	Command substitution system.	
roll CPU time	4-4	See CSS.	
signoff data and time	4-4	CONBFSZ	2-7
signon time left	4-5	Concurrent users	2-16
SVC CPU time	4-4	per accounting user class	2-32
task elapsed time	4-4	Console buffers	2-7
total processor time	4-4		2-10
total task elapsed time	4-4	Count of I/O transfers	4-10
user APU time	4-4	CREATE mode	3-5
user CPU time	4-4		3-7
Accounting transaction		CSBFSZ	2-7
files. See ATF.		CSS	
ACCOUNTS command	3-8	levels	2-14
ADD command	3-10		
ADD parameter	5-5	D	
Assistant task	5-3	Data collection	4-1
ATF	4-2		4-3
.MTM command	5-7	Data structures	2-34
parameter	5-7	Date and time submitted	4-6
AUF	3-1	Date formats	4-3
building and starting	3-4	Default volume	2-20
CREATE mode	3-5	DELETE command	3-14
parameter	5-2	Dynamic scheduling	2-24
REPORT mode	3-5		
START command	3-5	E	
UPDATE mode	3-5	Elapsed time	4-5
Authorized user file. See		END command	3-15
AUF.		End of task code	4-11
Authorized User Utility	3-1		5-11
Authorized User Utility		Error messages	6-13
commands		Error termination	6-4
ACCOUNTS	3-6		
ADD	3-8	F	
	3-3	File descriptor, task loaded	
	3-10	from	4-11
CHANGE	3-3	File descriptors	1-3
	3-12	Forced roll	2-23
DELETE	3-14		
END	3-15		
LIST	3-16		
MODE	3-18		
PAUSE	3-19		
PRIVILEGE	3-20		
TOTAL	3-26		

G		MTM command	
Global variables	2-27	format	5-4
new global	2-28	.MTM command	
new internal	2-28	examples	5-7
H		parameters	5-5
HASP interface	2-29	.MTM LLE commands	6-5
I		CPU	6-6
I/O		DISPLAY	6-2
class table	2-8	ELIGIBILITY	6-6
control block	2-9	examples	6-8
Input/output. See I/O.		LPU	6-6
INQUIRE parameter	5-7	OFF	6-6
Internal variables	2-13	ON	6-7
	2-28	PSM	6-7
IOTAB macro	2-8	.MTM LLE ELIGIBILITY	
J,K		dynamic queue 0	6-7
Job number	4-6	eligibility mode	6-7
	4-11	fixed queue 0	6-7
		eligibility mode	6-7
		nooperation mode	6-7
		.MTM PSM commands	6-5
		error messages	6-13
		I/O	6-12
		INTERNAL	6-12
		MAXPROCESSOR	6-12
		MTM START command	5-2
		MTM sysgen options	2-5
		batch job pause	2-20
		batch job PRC limit	2-16
		batch job priority	2-21
		command buffer size	2-7
		console buffer size	2-7
		default user volume	2-20
		forced roll	2-23
		global and local	
		variables	2-27
		HASP interface	2-29
		I/O class table	2-8
		I/O count limit per user	
		class	2-29
		MTM link	2-9
		new global variables	2-28
		new internal variables	2-28
		number of command buffers	2-12
		number of concurrent	
		users	2-16
		number of console buffers	2-10
		number of CSS levels	2-14
		number of message buffers	2-14
		number of sector length	
		buffers	2-15
		OS/32 accounting facility	2-19
		PRC limit per user class	2-32
		prompt device	2-17
		prompt sequence	2-17
		scheduling	2-23
		sector buffer size	2-6
		SPL/32 interface	2-27
		task queue size	2-19
		version identification	2-33
		MTMMAIN	2-1
L			
LINK OPTION command	2-9		
Link OPTION command	4-3		
LINKOPT	2-9		
LIST command	3-16		
LLE			
.MTM LLE command	6-6		
APU task queue assignment	6-1		
control of MTM subtasks	6-7		
error messages	6-7		
operating tips	6-10		
Load-leveling executive.	6-14		
See LLE.			
Local variables	2-27		
Logical unit	2-9		
M			
Memory requirements	2-34		
Memory used by a task	4-10		
Message buffers	2-14		
MESSAGE parameter	5-5		
MODE command	3-19		
MTM			
data structures	2-34		
description	1-1		
establishing accounts	3-1		
link	2-9		
operating procedures	5-1		
operation in a Model			
3200MPS System	6-1		
sysgen	2-1		

.MTMMASST TASK	5-3	PRMTSEQ macro	2-17
MTMPARMS.MAC	2-2	Processor time limit	3-3
MTMSGN.CSS	2-4	per account	3-4
MTMSTRUC.MLB macro library	2-4	Program development feature	2-13
Multi-terminal monitor. See MTM.		Prompt device	2-17
Multiprocessing environment	6-1	Prompt sequences	2-17
		PSM	2-23
		interval	6-1
N		PURGE parameter	6-3
			5-7
NCONBFS	2-10		
NCSBFS	2-12	Q	
NCSLVS	2-14		
NMSG	2-14	QUESIZE	2-19
NSECTS	2-15	Queue 0 eligibility	
Number of commands entered	4-6	algorithm	6-2
Number of CSS calls executed	4-6	fixed eligibility level	6-3
Number of DISPLAY FILES		QUIESCE parameter	5-6
commands	4-7		
Number of tasks executed	4-6	R	
NUMERS	2-16	REMOVE parameter	5-6
		Renamed and deleted files	4-7
O		REPORT mode	3-5
Operating procedures			3-7
end of task codes	5-12	Reporting Utility	4-1
error message	5-13	Roll count	4-10
loading and establishing		Roll time	4-9
MTM	5-1		
MTM START command	5-2	S	
Operation of the LIE		Secondary directory	2-6
computation-intensive		Sector buffer	2-6
task	6-2	SGN.\$HSP	2-29
CPU ready queue	6-2	SGN.@GLO	2-28
execution of MTM subtasks	6-2	SGN.@INT	2-28
queue 0	6-2	SGN.ACT	2-19
service-intensive task	6-2	SGN.BTCP	2-20
Optimization and control of		SGN.MVOL	2-20
load-leveling	6-4	SGN.PRIO	2-21
OS/32 accounting facility	2-19	SGN.ROLL	2-23
		SGN.SPL	2-27
P		SGN.VAR	2-27
PAGE parameter	5-3	Signoff time	4-12
PAUSE command	3-20	Signon and signoff date and	
PRC time	4-8	time	4-6
PRCLIMIT		Signon time limit	3-3
batch job	2-16	per account	3-4
I/O count limits	2-29	SPL/32 interface	2-27
job pause	2-20	new	2-28
per user class	2-32	START command	
relative default priority	2-21	Authorized User Utility	3-5
Prevention of CPU overload	6-3	MTM	5-2
Priority scheduling		STATUS parameter	5-5
dynamic	2-23	SUBMIT parameter	5-7
Priority scheduling		Subtask scheduling	
mechanism. See PSM.		dynamic	2-24
PRIVILEGE command	3-20	priority	2-24
Privileges		PSM	2-24
account	3-21	Subtasks of MTM	1-1
account 255	3-11	Sysgen	
default	3-11		
PRMTDEV macro	2-17		

MTM	4-2
OS/32	1-3
parameter file	4-2
Sysgen options	2-2
MTM	2-1
OS/32	2-5
SYSSTRUC.MLB macro library	1-3
System generation. See sysgen.	2-4
System requirements	1-2
	4-2

T

Task counter overflow	4-12
Task queue	2-19
Task termination	4-9
	4-12
TELL parameter	5-5
TERMINAL parameter	5-5
MTM START command	5-2
Terminal prompt	
device name	2-17
sequence	2-17
TERMINATE parameter	5-6
Time formats	4-3
TOTAL command	3-26

U

UIOLIM macro	2-29
UPDATE mode	3-5
	3-7
UPRCLIM macro	2-32
USED parameter	5-5
User identifier	4-7
	4-9
User-supplied information	4-11

V

Variables	
global	2-27
local	2-27
VERSION macro	2-33

W, X, Y, Z

WAIT parameter	5-5
Wait time	4-9

PERKIN-ELMER

PUBLICATION COMMENT FORM

We try to make our publications easy to understand and free of errors. Our users are an integral source of information for improving future revisions. Please use this postage paid form to send us comments, corrections, suggestions, etc.

1. Publication number _____

2. Title of publication _____

3. Describe, providing page numbers, any technical errors you found. Attach additional sheet if necessary.

4. Was the publication easy to understand? If no, why not?

5. Were illustrations adequate? _____

6. What additions or deletions would you suggest? _____

7. Other comments: _____

From _____ Date _____

Position/Title _____

Company _____

Address _____

STAPLE

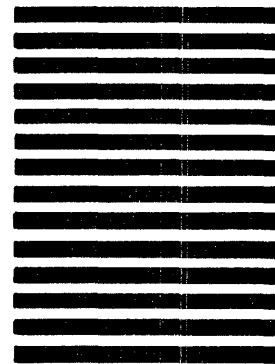
STAPLE

FOLD

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

PERKIN-ELMER

Data Systems Group
106 Apple Street
Tinton Falls, NJ 07724

**ATTN:
TECHNICAL SYSTEMS PUBLICATIONS DEPT.**

FOLD

FOLD

STAPLE

STAPLE