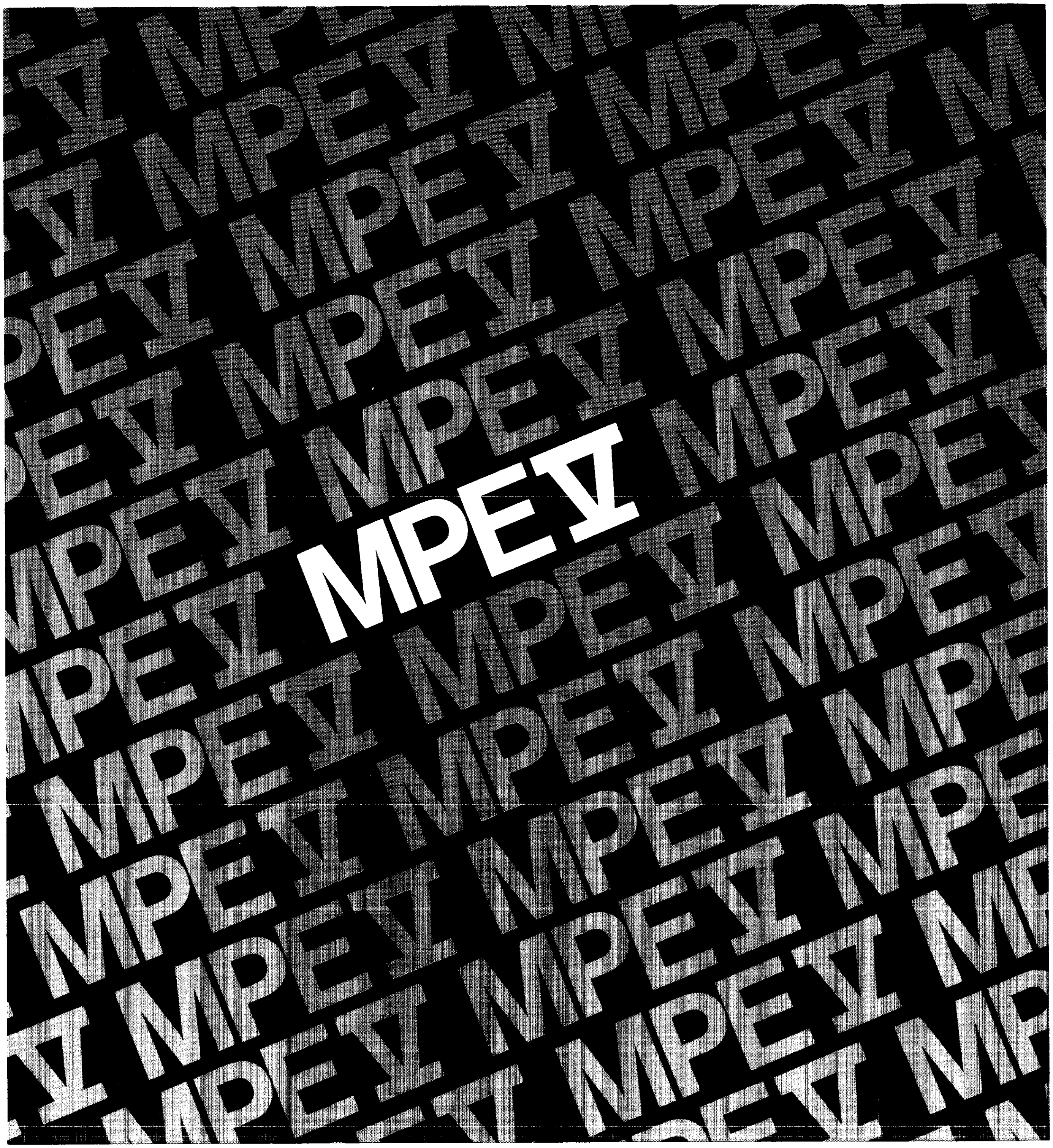
HP 3000 Computer Systems

HEWLETT PACKARD

MPE V Tables Manual for MPE V/E, Version G.00.00



HP 3000 Computer Systems

MPE V TABLES MANUAL
for MPE V/E, Version G.00.00

HEWLETT PACKARD

19447 PRUNERIDGE AVENUE, CUPERTINO, CA 95014

Part No. 32033-90010
E2412

Printed in U.S.A. 09/84

CAUTION

The normal checks and limitations that apply to the standard MPE users are bypassed in Privileged Mode. It is possible for a Privileged Mode program to destroy file integrity including the MPE operating system software itself. Upon request Hewlett-Packard will investigate and attempt to resolve problems resulting from the use of Privileged Mode code. This service is available on a time and materials billing basis. However, Hewlett-Packard will not support, correct, or attend to any modifications of the MPE operating system software.

NOTICE

The information contained in this manual is proprietary and may not be disclosed to third parties. The terms and conditions of release of this information to the recipient are governed by the agreement executed at and between Hewlett-Packard and the recipient regarding the HP 3000 internal MPE V System Table Structures, including the MPE V Tables Manual for MPE V Operating System.

Hewlett-Packard does not warrant the accuracy or completeness of the information contained in this manual. Hewlett-Packard may make revisions to the MPE operating system or CPU firmware that may affect the information and specifications contained herein. Hewlett-Packard shall have no obligation to provide the recipient with notification of revisions, corrections, or updates regarding the information contained herein.

Hewlett-Packard shall not be responsible for any loss or damage to the recipient, its customers, or any third party caused by the recipient's use of the information provided herein.

Copyright (c) 1984 by HEWLETT-PACKARD Company

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the current edition, and lists the dates of all changed pages. Unchanged pages are listed as "ORIGINAL". Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. Changes are marked with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars and dates remain. No information is incorporated into a reprinting unless it appears as a prior update.

First Edition . . . . September 1984

Effective Pages	**Date**
ALL	SEP 1984

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page and back cover of the manual changes only when a new edition is published. When an edition is reprinted, all the prior updates to the edition are incorporated. No information is incorporated into a reprinting unless it appears as a prior update.

The software date code number printed alongside the date indicates the version level of the software product at the time the manual edition or update was issued. Many product updates and fixes do not require manual changes, and conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

First Edition . . . . SEP 1984 . . . . G.00.00

CONTENTS

PREFACE xix

CHAPTER 1 MEMORY LAYOUT

Fixed Low Memory (Series 44/48/64/68) 1-1
System Global Area 1-3
SysGlob Extension 1-14
SYSDB Words 1-17
SysGlob Word Definitions 1-17
Allow Mask Format 1-18
Logging Related Locations. 1-20
FLAGX 1-20
Process Stop List General Layout 1-21
  Entry Format 1-21
  Preassigned Entries 1-21
Initial Memory Allocation 1-22
Bank 0 1-23
Bank 1 1-24

CHAPTER 2 MEMORY MANAGEMENT TABLES

Segment Table Structure. 2-1
  Pointers and DST #'s of Segment Table Components 2-2
  Standard Object Identifier Format 2-3
  DST Entry Formats 2-3
  CST Entry Formats 2-5
  ST Entry Field Descriptions. 2-6
  CSTBLK Format 2-6
Program Blocks and the CSTXMAP 2-7
  CST Extension Block Entry Format. 2-7
  Fixed DST Entry Assignments 2-8
Swap Tables 2-11
  SWAPTAB Entry 0 Format. 2-11
  SWAPTAB Unassigned Entry Format. 2-12
  Segment Locality Lists (SLL) 2-13
  SLL Header Format. 2-14
  SLL List Entry Format 2-15
  Special Request Table. 2-16
Main Memory Region Headers and Trailers 2-18
  Global Region Trailer. 2-19
  Global Region Header (Available Regions) 2-19
  Subregion Header (Available Regions) 2-20
  Global Region Header (Reserved Regions) 2-21
  Subregion Header (Reserved Regions) 2-22
  Subregion Header (Cached Regions) 2-23
  Region Header and Trailer Field Descriptions 2-24
  Space Allocation Structures. 2-26

CHAPTER 3 DISC LAYOUT

System Disc Layout 3-1
Disc Label (Sector 0 of Disc) 3-4

CONTENTS (Continued)

System Volume 3-4
Serial Volume 3-6
Master Volume 3-8
Slave Volume 3-10
Defective Tracks Table (Sector 1 of Disc) 3-12
Defective Sector Table (DSCT -- Sector 1 of Disc) 3-13
Reserved Area Bit Map (Sector 4 of the System Disc) 3-14
Disc Cold Load Information Table (Sectors 28-30) 3-15
INITIAL Program CST Map 3-20
SYSDUMP/INITIAL Communication Record 3-21
Cold Load Information Table Extension 3-23
Virtual Disc Space Management Structures 3-24
Virtual Disc Space Management Table 3-24
General Structure 3-24
VDSMTAB Entry 0 Format 3-25
VDSMTAB General Entry Format 3-26
Volume Table 3-27
Typical Private Volume Entry 3-28
Typical System Volume Entry 3-29

CHAPTER 4 DIRECTORY

Introduction to the Directory 4-1
Overview of Directory 4-2
Directory Data Segment 4-3
Directory Pointer Area [DA or DB] DST=20(10) 4-5
Directory Space Data Segment (DIRSDS) 4-6
Directory Structure 4-10
Directory Definitions 4-11
Index Block Prefix (10 Words) 4-11
Index Entry (6 Words) 4-12
Account Entry (%36 Words) 4-12
Group Entry (51% Words) 4-14
File Entry (File Pointer) (6 Words) 4-16
User Entry (19 Words) 4-17
User Attributes/Capabilities 4-18
Volume Set Definition Entry 4-19
GVSLINKAGE 4-20
GVSINFO 4-20
GVSVOLFLAGS 4-20
GVSVOLINFO 4-20
Volume Set Class Entry 4-21
GVCLINKAGE 4-22
GVCINFO 4-22
Volume Mask Format 4-22

CHAPTER 5 LOCK RESOURCES

SIR# Allocation DST %53 5-1
Sir's Ordered by Sir Number 5-1
Sir's Ordered by Ranking 5-2

SIR Table Information 5-3
SIR Entry Formats 5-4
RIN Table General Layout (Initialized State) 5-5
Allocation and Locking of Local RINS 5-6
Allocation and Locking of File RINS 5-7
Allocation and Locking of Global RINS 5-8

CHAPTER 6 FILE SYSTEM

File System Overview 6-1
Buffers 6-2
Table Formats 6-3
File System Section of PCBX (PXFILE) 6-3
Overhead 6-4
PXFILE Control Block Table (PXFCBT) 6-6
Available Block 6-6
Active File Table (AFT) 6-7
File Control Block Table (CBTAB) 6-2
Overhead 6-13
Vector Table 6-15
Control Block Area 6-17
Access Control Block (ACB) 6-18
Logical Access Control Block (LACB) 6-19
Physical Access Control Block (PACB) 6-21
File Control Block (FCB) 6-37
File Label (FLAB) 6-43
File Multi-Access Vector Table (FMAVT) DST(%54) 6-52
Zero Entry Format 6-52
Typical Entry Format 6-53
System Global Area (SYSGLOB) 6-54
SIRs, Locks, and Deadlocks 6-54
Shared CBT DST 6-55

CHAPTER 7 PROCESS TABLES

Process Control Block Table Structure and Format 7-1
Fixed Cells Related to PCB 7-1
PCB Entry 0 Format 7-2
Unassigned PCB Entry Format 7-3
Assigned PCB Entry Format 7-4
PCBX Structure and Format 7-8
PCBX General Structure 7-8
PXGLOB Format 7-9
PXFIXED Assignments 7-10
PXFIXED Expansion Bitmap 7-14
PCBX For Core Resident System Process Stacks 7-15
Process To Process Communication Table 7-16
Subsystem Reserved DL Area 7-17
FORTRAN Logical Unit Table (FLUT) 7-18

CONTENTS (Continued)

CHAPTER 8 JOB TABLES

Job Tables Overview 8-1
Job Master Table Structure (JMAT) 8-2
Job Master Table Entry (JMAT) 8-4
Job States 8-6
Job Process Count Table (JPCNT) 8-7
Job Cutoff Table (JCUT) 8-8
Job Information Table (JIT) 8-9
Allow Mask Format 8-11
Job Directory Table (JDT) 8-14
  Job Data Segment Directory Entry (In JDT) 8-15
  Job Temporary File Entry (In JDT) 8-15
  File Equation Table Entry (In JDT) 8-16
  Job Line Equation Entry 8-17
Job Control Word Table (JJCW) 8-18
Aoptions and Foptions Word Breakdown 8-19
PMASK Word Breakdown 8-20
UCOP Request Queue (DST#9) 8-21
UCOP Entry Format 8-22

CHAPTER 9 RELOCATABLE OBJECT CODE

USL Files Introduction 9-1
Record 0 and Overall USL File Format 9-1
Data Descriptors, Passed Parameters 9-5
  Pascal 9-5
  Entry Type 0 9-6
  Entry Type 1 9-6
  Entry Type 2 9-8
  Entry Type 3 9-11
  Entry Type 4 9-11
  Entry Type 5 9-14
  Entry Type 6 9-15
  Entry Type 7 9-17
  Entry Type 8 9-19
Entry Header Format 9-21
  Header Type 0 9-22
  Header Type 1 9-22
  Header Type 2 9-23
  Header Type 3 9-23
  Header Type 4 9-24
  Header Type 5 9-24
  Header Type 6 9-25
  Header Type 7 9-25
  Header Type 8 9-26
  Header Type 9 9-27
  Header Type 10 9-29
  Header Type 11 9-29
RL File Format 9-30
Storage Management 9-31

Entry Point Directory 9-32
Typical Directory Entry 9-33
Procedure Information Block 9-34
Headers 9-35

CHAPTER 10 PREPARED OBJECT CODE

Program File Format 10-1
Flags 10-3
Flags2 10-4
CST Remapping Array. 10-4
Segment Descriptor Array 10-4
Global Area Format 10-4
External List. 10-5
Entry Point List 10-6
Code Segment With Patch Area 10-7
Patch Area 10-7
PMAP Information 10-8
PMAP Type Table 10-8
PMAP Records 10-9
Type 0 Segment PMAP Record 10-9
Type 1 Procedure PMAP Record 10-9
Type 2 Secondary Entry PMAP Record 10-10
SL File Format. 10-11
Storage Management. 10-13
Entry Point Directory 10-13
Typical Directory Entry 10-14
Code Segment Linkage Structure 10-15
Reference Table Structure. 10-17
Reference Table (256 Maximum Entries) 10-18
Code Segment With Patch Area 10-19
PMAP Information 10-20
PMAP Type Table 10-20
PMAP Records 10-21
Type 0 10-21
Type 1 10-21
Type 2 10-21

CHAPTER 11 LOADER

MPE Loader 11-1
Loader Segment Table Overview 11-1
LST Overview. 11-1
XLST Overiew 11-2
Loader Segment Table Primary DB 11-3
Directory Entries 11-4
Loader Cache 11-10
Cache Data Segment Format 11-10
Bucket Format 11-10
Loader Communication Table (LCT) 11-11

CONTENTS (Continued)

Form Incoming to Loader (Load/Allocate Program) 11-11
Form Incoming to Loader (Load/Allocate Procedure) 11-12
Form Returned (No Error) 11-13
Form Returned (Error Occurred) 11-13
Logical Segment Transform Table (LSTT) 11-14

CHAPTER 12 PRIVATE VOLUMES / SERIAL DISC

Mounted Volume Table (MVT AB) 12-1
Private Volume User Table (PVUSER)) 12-4
Bind Names Data Segment 12-6
Serial Disc Tables and Data Structures 12-8
Data Record Format 12-8
End of File Format 12-9
Contiguous Block Format 12-10
Hole Format 12-10
Gap Table Format 12-11
SDISC Extra Data Segments 12-13
Serial Disc Organization 12-15

CHAPTER 13 I/O

I/O Table Linkage 13-1
Device Reference Table (DRT) 13-2
Driver Linkage Table (DLT.) 13-3
Logical-To-Physical Device Table (LPDT) 13-4
Entry 0 13-5
Typical Entry (Virtual Devices) 13-5
Typical Entry (All Real Devices) 13-6
Entry for Terminal-Like Devices 13-7
Entry for Tape Drives 13-7
Entry for Disc Drives 13-8
Logical Device Table (LDT) 13-9
Overview of Data Segment 13-9
Zero Entry Format 13-9
Typical Entry Format 13-10
Logical Device Table Extension (LDTX) 13-11
Overview of Data Segment 13-11
Zero Entry 13-12
Typical Entry 13-12
Terminal Entry 13-13
Serial or Foreign Disc Entry 13-14
CIPER Entry 13-14
System or Private Volume Disc Entry 13-14
Device Class Table (DCT) 13-15
Overview of Data Segment 13-15
Header Entry Format 13-15
Typical Entry Format 13-16
Discussion 13-16
Interrupt Linkage Table (ILT) 13-18

Device Information Table (DIT) 13-20
DIT for HP-IB Systems 13-20
DIT Terminology for HP-IB Systems 13-21
Device Information Table (DIT) for CIPER 13-21
DIT for Channel Devices 13-24
DIT For 7905/7906/7920/7925 13-26
Error and Retry Information 13-28
CS 80 Disc Device Information Table (DIT) 13-29
DIT For 7970 Magnetic Tape 13-32
DIT for 7976 Magnetic Tape 13-34
Card Reader DIT 13-37
Card Reader DIT Field Definitions 13-38
Device Information Table for HP-IB Card Reader 13-39
2608 Line Printer DIT (HP-IB Systems) 13-41
2608 Line Printer Status 13-43
2631 or 2619A Line Printer DIT (for HP-IB Systems) 13-44
2680A/2688A DIT 13-46
I/O Status Block 13-48
Disc Request Table and Disc Requests 13-51
DISCREQTAB 13-51
Disc Request Table 13-52
Disc Request Table Entry 0 Format 13-52
Disc Request Element Format 13-53
IOQ Table Layout 13-56
I/O Queue Element (IOQ) 13-58
I/O System Status Returns 13-60
I/O Queue Element for 7976A Magnetic Tape 13-62
I/O Queue Element (IOQ) for CIPER 13-64
HP-IB CIPER Physical Driver Request Codes 13-65
CIPER Driver Return Status Codes 13-65
2608 Line Printer I/O Queue Element (HP-IB Systems) 13-66
2608 Line Printer Request Codes 13-69
2619A & 2631 Line Printer IOQ Element(HP-IB Systems) 13-70
2619 Line Printer Request Codes 13-73
2631 Line Printer Request Codes (HP-IB Systems) 13-74
I/O Queue Element For HP-IB Card Reader 13-75
CS 80 Disc Request Queue Element (IOQ) 13-77
CS 80 Integrated Cartridge Tape Request 13-80
SBUF Table Layout 13-83
Table Element Allocation (SBUF) 13-84
ICS Global 13-86
ICS Global Cells With Initial Values 13-89
CS 80 Disc Interrupt Linkage Table (ILT) 13-90

CHAPTER 14 SPOOLING

Input Device Directory/Output Device Directory 14-1
Overview of Table Structure 14-1
Entry 0 (Overall Table Definitions) 14-2

CONTENTS (Continued)

Typical Head Entry (4 words) 14-3
Typical Subentry (%40 words) 14-4
SPOOK Tape Format 14-6
Label Record 14-6
File Directory. 14-7
Device and Class Directory 14-7
Logical Device Entry 14-7
Device Class Entry 14-8
Spoolfile Format 14-8
Spoolfile Block Format 14-8
Spoolfile Record Format 14-8
User Labels Information 14-9

CHAPTER 15 UNIFIED COMMAND LANGUAGE (UNCL)

Reply Information Table (RIT) 15-1
Message System General Description 15-2
Message Catalog. 15-3
MAKECAT Program 15-4
Message System CATALOG.PUB.SYS 15-5
Message Set Directory. 15-6
HELP Subsystem. 15-7
UDC Directory 15-8
UDC's COMMAND.PUB.SYS 15-9
CI Stack Definition 15-11
Field Definitions 15-12
Association DST Layout 15-13

CHAPTER 16 SYSDUMP/INITIAL

CONFDATA File 16-1
Record 0 of CONFDATA File (CTAB0) 16-1
Record 1 of CONFDATA File (CTAB) 16-2
DEVDATA.PUB.SYS 16-4
Overview 16-4
Parameter Record. 16-4
Driver Table 16-6
SYSDUMP Format. 16-7
WSC Table Format 16-9
Series 64/68 WCS Table Format 16-9
Store Tape Format. 16-10
First Volume 16-10
Subsequent Volumes 16-12
End of Volume 16-13

CHAPTER 17 MISCELLANEOUS

Labeled Tape Subsystem 17-1
Tape Label Table 17-5

Volume Recognition 17-10
Opening a File 17-10
Reading and Writing Files 17-10
Closing Files 17-11
Store-Restore 17-11
Miscellaneous 17-11
Breakpoint Table 17-12
PCB Breakpoint Extension Table .17-13
Breakpoint Entry Table 17-13
Active Entry 17-14
Timer Request List (TRL) 17-17
MPE User Logging 17-19
General Design Overview 17-19
Hardware Environment. 17-19
Software Environment 17-19
Design Narrative 17-19
Error Recovery Description. 17-20
Design Structures 17-21
User Logging Table 17-21
User Logging Buffer 17-27
User Logging Identifier Table. 17-37
Logging Record Format. 17-41
Measurement Information Table 17-44

CHAPTER 18 MESSAGE FILES

Message File Data Structures 18-1
File Structure 18-1
Block Structure 18-2
Record Format 18-2
Header Format 18-3
Message Access Control Block. 18-3
MMSTAT Definitions. 18-9
File System Basic IPC Definitions. 18-11
General Behavior 18-11
Port Data Structures 18-13
Port Data Segment. 18-13
Port With Two Outstanding Messages 18-13
Port Number. 18-14
Port DST Number Array. 18-14
Port Data Segment Global Area 18-15
Port 18-16
Message Queue Entry (MQE) 18-17
File System Message Files 18-17
Timer List Entry (TLE). 18-18
MMSTAT Definitions. 18-18

CHAPTER 19 MPE MEMORY RESIDENT MESSAGE FACILITY

Overview of Facility. 19-1
Message Intrinsics 19-1

CONTENTS (Continued)

SENDMSG 19-1
PORTSTATUS 19-2
RECEIVEMSG 19-2
Supporting Data Structures 19-3

CHAPTER 20 MMSTATS EVENTS

MMSTATS Catalog Index 20-1
MMSTAT Event Group 0 (Memory Management Events) 20-4
Event 0 20-4
Event 1 20-5
Event 2 20-6
Event 4 20-7
Event 5 20-8
Event 6 20-9
Event 7 (%7) 20-9
Event 8 (%10). 20-10
MMSTAT Event Group 1 (Memory Manager) 20-11
Event 12 (%14) 20-11
Event 13 (%15) 20-11
Event 14 (%16) 20-12
Event 15 (%17) 20-12
Event 16 (%20) 20-13
Event 17 (%21) 20-13
Event 18 (%22) 20-13
MMSTAT Event Group 2 20-14
Event -20 (-%24). 20-14
Event -21 (%25) 20-14
Event -23 (-%27). 20-15
MMMSTAT Event Group 3 20-15
MMSTAT Event Group 4 (Scheduling) .20-16
Event 40 (%50) 20-16
MMMSTAT Event Group 5 20-17
MMSTAT Event Group 6 (FILESYS) 20-18
Event -60(%74). 20-18
Event -61(%75). 20-19
Event -60(%74). 20-19
Event -61(%75). 20-20
Event -62(%76). 20-21
Event -63(%77). 20-21
Event -64(%100) 20-22
Event -65(%101) 20-23
Event -66(%102) 20-24
Event -67(%103) 20-24
Event -68(%104) 20-25
Event -69 (%105). 20-25
MMSTAT Event Group 7 (FILESYS) 20-26
Event -70 (%106). 20-26
Event -71 (%107). 20-26
Event -72 (%110). 20-27

Event -74 (%112). 20-27
Event -75 (%113). 20-28
Event -76 (%114). 20-28
Event -77 (%115). 20-29
Event -78 (%116). 20-29
Event -79 (%117). 20-30
MMSTAT Event Group 8 20-31
Event -80 (%120). 20-31
Event -81 (%121). 20-31
Event 83 (%123) 20-32
Event 84 (%124) 20-33
Event 86 (%126) 20-33
Event 87 (%127) 20-33
Event 88 (%130) 20-34
Event 89 (%131) 20-34
MMSTAT Event Group 9 (Disc I/O Requests) 20-35
Event 90 (%132) 20-35
Event -98 (%142). 20-35
MMSTAT Event Group 10 20-36
Event 100 (%144). 20-36
Event 101 (%145). 20-36
MMSTAT Event Group 11 20-37
Event -110 (%156) 20-37
Event -111 (%157) 20-37
MMSTAT Event Group 12 20-38
Event 120 (%170). 20-38
Event 125 (%175). 20-38
MMSTAT Event Group 13 20-39
Event 139 (%213) 20-39
MMSTAT Event Group 14 (CS/3000) 20-40
Event 140 (%214). 20-40
Event 142 (%216). 20-41
Event 144 (%220). 20-42
Event 146 (%222). 20-42
Event 147 (%223). 20-43
Event 149 (%225). 20-43
MMSTAT Event Group 15 (CS/3000) 20-44
Event 150 (%226). 20-44
Event 152 (%230). 20-44
Event 153 (%231). 20-45
Event 154 (%232). 20-45
Event 155 (%233). 20-46
MMSTAT Event Group 16 20-47
Event 160 (%240). 20-47
MMSTAT Event Group 19 20-48
Event 191 (%277). 20-48
Event 192 (%300). 20-49
Event 193 (%301). 20-50
Event 194 (%302). 20-51
Event 195 (%303). 20-51

CONTENTS (Continued)

MMSTAT Event Group 20 20-52

Event 200 (%310). 20-52

Event 201 (%311). 20-52

MMSTAT Event Group 21 Process Creations and Terminations

Logical Process 20-53

Logical Process Table 20-53

Event -211 (%323) 20-53

MMSTAT Event Group 22 20-54

Time Stamp of Event Trace Enable and Disable 20-54

Event 221 (%335). 20-54

Event 222 (%336). 20-55

Event -223 (-%337) 20-55

Event -224 -(%340) 20-56

Event -225 (-%341) 20-56

Event -226 (-%342) 20-56

Event -227 (-%343) 20-57

Event -228 (%344) 20-57

Event -229 (-%345) 20-57

MMSTAT Event Group 23 (Terminal I/O) 20-58

Event 230 (%346). 20-58

Event 231 (%347). 20-58

Event 232 (%350). 20-59

Event 233 (%351). 20-59

Event 234 (%352). 20-60

Event 235 (%353) 20-60

Event 236 (%354). 20-61

Event 237 (%355). 20-61

Event 238 (%356). 20-61

MMSTAT Event Group 24 (Power Fail) 20-62

Event 240 (%360). 20-62

CHAPTER 21 ROOTFILE LAYOUT

General Rootfile Layout. 21-1

Root File Label 0 21-2

Root File Labels 1 & 2 21-5

Root File Label 3 21-6

Root File- Next Label 21-7

Item/Set Read/Write Table Format 21-8

Root File Record 0 21-9

Root File Record 1 21-10

Root File- Next Record. 21-11

Data Set Control Blocks (DSCB)- General Layout 21-12

Data Set Control Block (Global Area) 21-13

Data Set Control Block (Item Numbers) 21-14

Data Set Control Block (Record Definition Item Displacement) 21-14

Data Set Control Block (Path Table) 21-15

General Data Set Layout 21-15

Data Set User Label 0. 21-16

Data Set Records 21-16

CHAPTER 22 DISC FREE SPACE MAP

Disc Resident Data Structures 22-1
Bit Map 22-1
Descriptor Table (DT) 22-1
Virtual Memory Resident Data Structures 22-2
Disc Free Space Data Segment 22-2

CHAPTER 23 MPE DISC CACHING

Disc Caching Overview 23-1
Disc Caching Tables Overview. 23-4
Cache Directory Table. 23-6
Header Entry 23-7
Device Entry 23-10
Mapped Domain Entry 23-13
Logical Disc Request Table 23-16
Logical Disc Request Entry 23-17

PREFACE

This manual describes the internal table organization of the MPE V operating system, release G.00.00. The Tables Manual is an informational reference for the technically sophisticated user with Privilege Mode capability. We strongly discourage modifying the table structure because you may destroy the operating system. The following caution applies:

CAUTION

> The normal checks and limitations that apply to the standard MPE users are bypassed in Privileged Mode. It is possible for a Privileged Mode program to destroy file integrity including the MPE operating system software itself. Upon request Hewlett-Packard will investigate and attempt to resolve problems resulting from the use of Privileged Mode code. This service is available on a time and materials billing basis. However, Hewlett-Packard will not support, correct, or attend to any modifications of the MPE operating system software.

The table structure of MPE V is significantly expanded from MPE IV. The operating system reflected in the table structure is the Fundamental Operating Software (FOS) version of MPE V. Your table structure may look different depending on the applications and uses of your system.

The information is presented in several different formats. This reflects the combined knowledge of several divisions and groups within Hewlett-Packard. Instead of taking the time to consolidate all the various formats, we chose to release the information quickly.

We hope you will find this edition informative. Your comments and suggestions are welcome via the "Reader Comment Sheet" at the back of this manual.

CHAPTER 1 MEMORY LAYOUT

Fixed Low Memory (Series 44/48/64/68)

```
 %---------------------------------------------DEC
 0|     CSTB (BASE OF CST TABLE)**             |0
  ---------------------------------------------
 1|     XCSTB (POINTER TO CURRENT EXECUTING    |1
                PROGRAM BLOCK)
  ---------------------------------------------
 2|     DSTB (BASE OF DST TABLE)**             |2
  ---------------------------------------------
 3|                    0                       |3
  ---------------------------------------------
 4|     CPCB (CURRENT PCB INDEX  )**           |4 >PCB REL
  ---------------------------------------------
 5|     QI (INITIAL Q FOR ICS)**               |5
  ---------------------------------------------
 6|     ZI (INITIAL Z FOR ICS)**               |6
  ---------------------------------------------
 7|     SYSTEM INTERRUPT MASK WORD**           |7
  ---------------------------------------------
10|     DRTBANK (BANK OF DRT TABLE)            |8
  ---------------------------------------------
11|     DRTADDR (BASE OF DRT TABLE)            |9
  ---------------------------------------------
12|     DBBANK (FOR INITIAL'S STACK) *         |10
  ---------------------------------------------
13|     DB (FOR INITIAL'S STACK)     *         |11
  ---------------------------------------------
14|                                            |12
  ---------------------------------------------
15|                                            |13
  ---------------------------------------------
16|                                            |14
  ---------------------------------------------
17|                                            |15
  ---------------------------------------------
20|                                            |16
  ---------------------------------------------
21|     LR (INTERRUPT INTERVAL)+               |17
  ---------------------------------------------
22| TEMPLR (TEMP STORAGE OF LIMIT REG)+       |18
  ---------------------------------------------
23|  LR   (SYSTEM CLOCK LIMIT REGISTER) **     |19
  ---------------------------------------------
24|////////////////////////////////////////////|20
  ---------------------------------------------
```

Fixed Low Memory (Series 44/48/64/68) (Cont.)

```
      ---------------------------------------------
   25| TR (TIME SINCE LAST SOFT TIMER INTERRUPT)** |21
      ---------------------------------------------
   26|     SCST (SYSTEM CLOCK STATUS)**            |22
      ------------  -------------------------------
   27|     SCLC (SYSTEM CLOCK LAST COUNT)**        |23
      ------ --------------------------------------
30-37|                                             |24-31
      ---------------------------------------------
```

NOTE: All pointers are absolute addresses.

```
LEGEND:  ** Needed by Firmware and/or by System, always
         *  Needed during INITIAL
         +  Needed by MPE, set up by INITIAL or PROGENITOR.
```

System Global Area

```
                                    1  1  1  1  1  1
OCTAL   1  2  3  4  5  6  7  8  9  0  1  2  3  4  5        NAME
-----                                                      ----
      |-----------------------------------------------|
     0|            SYSGLOB                            |
      |-----------------------------------------------|
     1|           CST BASE                            |CST
      |-----------------------------------------------|
     2|           DST BASE                            |DST
      |-----------------------------------------------|
     3|           PCB BASE                            |PCB
      |-----------------------------------------------|
     4|         SWAPTAB BASE                          |SLL
      |-----------------------------------------------|
     5|           IOQ BASE                            |IOQ
      |-----------------------------------------------|
     6|          SBUF BASE                            |BUF
      |-----------------------------------------------|
     7|             ICS QI                            |ICS
      |-----------------------------------------------|
    10|          LPDT BASE                            |LPDT
      |-----------------------------------------------|
    11|         SMON BASE                             |SMON
      |-----------------------------------------------|
    12|           TRL BASE                            |TRL
      |-----------------------------------------------|
    13|          JCUT BASE                            |SIR
      |-----------------------------------------------|
    14|           SIR BASE                            |SDCTAB
      |-----------------------------------------------|
    15|         JPCNT BASE                            |JPCNT
      |-----------------------------------------------|
    16|          TBUF BASE                            |BUF
      |-----------------------------------------------|
    17|        DISC REQUEST BASE                      |DRQ
      |-----------------------------------------------|
    20|                                               |
      |          FIRST FREE MEMORY ADDRESS            |
    21|                                               |
      |-----------------------------------------------|
    22|                                               |
      |             TIME OF LAST CYCLE                |
    23|                                               |
      |-----------------------------------------------|
    24|              RESERVED                         |
      |-----------------------------------------------|
    25|          Break Point Flag                     |BPTF
      |-----------------------------------------------|
```

System Global Area (Cont.)

```
  |-----------------------------------------------|
26|              VDSMTAB BASE                     |VDSMTAB
  |-----------------------------------------------|
27|             STATIC FENCE                      |
  |-----------------------------------------------|
30|             CURRENT CST BLOCK INDEX           |CSTBX
  |-----------------------------------------------|
31|             MEASIO BASE                       |MEASIO
  |-----------------------------------------------|
32|      DISPLACEMENT TO CODE =@CST(0)-@DST(0)    |DFC
  |-----------------------------------------------|
33| DISPLACEMENT TO SHARABLE = @CST(LAST)-@DST(0) |DFS
  |-----------------------------------------------|
34|          Smon Index                           |
  |-----------------------------------------------|
35|            ABS ADDRESS (SYSDIT(8))            |DIT8
  |-----------------------------------------------|
36|           Reserved                            |SBANK
  |-----------------------------------------------|
37|  ABS ADR OF PMBC TABLE FOR LST/STT CHECKING   |SBASE
  |-----------------------------------------------|
40|  RESERVED FOR INITIAL (VDSENTRY)              |
  |-----------------------------------------------|
41|  RESERVED FOR INITIAL (VDSMAP)                |
  |-----------------------------------------------|
42|           SRTTAB BASE                         |SRTTAB
  |-----------------------------------------------|
43|           SPECQ HEAD                          |SPECQHEAD
  |-----------------------------------------------|
44|   Number of Available Regions                 |HOLECOUNT
  |-----------------------------------------------|
45|  # PAGES IN LARGEST CURRENTLY AVAILABLE REGION|MAXAVAILREG
  |-----------------------------------------------|
46|  MAKE OVERLAY CANDIDATE INFORMATION           |MOCINFO
  |-----------------------------------------------|
47|     NUMBER OF MEMORY BANKS  CONFIGURED -1     |NBANKS
  |-----------------------------------------------|
50|        SCHEDULER TO AWAKE MESSAGE             |SCHEDTOAWAKEMSG
  |-----------------------------------------------|
51|            POINTER TO CSTBLK TABLE            |CSTXBLCKPOINTER
  |-----------------------------------------------|
52|        AWAKE TO SCHEDULER MESSAGE             |AWAKETOSCHEDMSG
  |-----------------------------------------------|
53|        WAIT TO SCHEDULER MESSAGE              |
  |-----------------------------------------------|
54|        CURRENT ACTIVITY'S PRIORITY            |CURACTPRI
  |-----------------------------------------------|
```

System Global Area (Cont.)

```
              |-------------------------------------------------|
          /55 |             BUSY TABLE POINTER                  |BUSY
          |   |-------------------------------------------------|
          |56 |             HEAD TABLE POINTER                  |HEAD
          |   |-------------------------------------------------|
          |57 |             TAIL TABLE POINTER                  |TAIL
          |   |-------------------------------------------------|
          |60 |         # OF SIO PROGRAMS EXECUTING             |SIOCOUNT
          |   |-------------------------------------------------|
          |61 |         PARITY ERROR FLAG (MEM PE)              |PARITY
          |   |-------------------------------------------------|
          |62 | Impeded queue head for message buffer (PIN)     |IOMSGPIN
          |   |-------------------------------------------------|
          |63 |        I/O Message system error flags           |IOLOGQX
              | (0:1) - No SYSBUF avail for I/O error logging   |
              | (1:1) - No SYSBUF for IOMESSAGE (GENMSG)        |
          |   |-------------------------------------------------|
RESERVED  |64 |         # OF TERMINALS READING                  |RDCOUNT
FOR I/O  <    |-------------------------------------------------|
SYSTEM    |65 |         # OF TERMINALS WRITING                  |WRTCOUNT
          |   |-------------------------------------------------|
          |66 |                DSET B                           |CRIO
          |   |-------------------------------------------------|
          |67 |                                                 |CRIO
          |   |              LAST TIMER                         |
          |70 |                                                 |CRIO
          |   |-------------------------------------------------|
          |71 |           HIGHEST DRT NUMBER                    |HSYSDRT
          |   |-------------------------------------------------|
          |72 |               POWERFAIL                         |POWERFAIL
          |   |-------------------------------------------------|
          |73 |             SYSTEM UP FLAG                      |SYSUP
          |   |-------------------------------------------------|
          \74 |     SYS CONSOLE LOGICAL DEVICE NUMBER           |CONSLDEV
              |-------------------------------------------------|
         / 75 |             COLD LOAD COUNT                     |CLOADID
         |    |-------------------------------------------------|
         | 76 |             SHARED FCB DST                      |SHFCBDST
         |    |-------------------------------------------------|
         | 77 |            MONITORING FLAGS                     |
         |    |-------------------------------------------------|
RESERVED |100 |                                                 |
FOR FILE<     |          MAX # OF SPOOL SECTORS                 |MAXSSECT
SYSTEM   |101 |                                                 |
         |    |-------------------------------------------------|
```

System Global Area (Cont.)

```
     |-------------------------------------------------|
|102 |                                                 |
|    |        CURRENT # OF SPOOL KILOSECTORS           |NUMSSECT
|103 |                                                 |
|    |-------------------------------------------------|
\104 |          # SECTOR/SPOOLFILE EXTENT              |EXTSSECT
     |-------------------------------------------------|
 105 |          MAX CODE SEGMENT SIZE                  |
     |-------------------------------------------------|
 106 |      MAX # OF CODE SEGMENTS/PROCESS             |
     |-------------------------------------------------|
 107 |          MAX STACK SIZE (MAXDATA)               |
     |-------------------------------------------------|
 110 |            DEFAULT STACK SIZE                   |
     |-------------------------------------------------|
 111 |        MAX EXTRA DATA SEGMENT SIZE              |
     |-------------------------------------------------|
 112 |    MAX # EXTRA DATA SEGMENTS/PROCESS            |
     |-------------------------------------------------|
 113 |       DST number for MESSAGE buffers            |
     |-------------------------------------------------|
 114 |              UPDATE LEVEL                       |UPDATEL
     |-------------------------------------------------|
 115 |               FIX LEVEL                         |FIXL
     |-------------------------------------------------|
 116 |             VERSION LEVEL                       |VERSION
     |-------------------------------------------------|
 117 |          DEFAULT CPU TIME LIMIT                 |
     |-------------------------------------------------|
 120 |          # OF SECONDS TO LOGON                  |
     |-------------------------------------------------|
 121 |           JOBSYNCH BITS (13:3)                  |
     |-------------------------------------------------|
 122 |        EXTERNAL PLABEL OF INITIATE              |
     |-------------------------------------------------|
 123 |        INTERNAL PLABEL OF INITIATE              |
     |-------------------------------------------------|
 124 |               MAXSYSDST                         |
     |-------------------------------------------------|
 125 |               MAXSYSCST                         |
     |-------------------------------------------------|
 126 | Ldev for SL.PUB.SYS   | HODA for SL.PUB.SYS     |
     |-------------------------------------------------|
 127 |           LODA for SL.PUB.SYS                   |
     |-------------------------------------------------|
 130 |              (DIRECTORY)                        |
     |                                                 |
 131 |             (DISC ADDRESS)                      |
     |-------------------------------------------------|
```

System Global Area (Cont.)

```
              |-------------------------------------------------|
          132 |                 SPOOLINDEX                      |
              |-------------------------------------------------|
         /133 |            EXT LABEL FOR SHOWCOM                |
         |    |-------------------------------------------------|
         |134 |                                                 |
         |    |-------------------------------------------------|
         |135 |              CS IOWAIT PLABEL                   |
RESERVED<     |-------------------------------------------------|
FOR CS   |136 |                       |        CS FIX LEVEL     |
         |    |-------------------------------------------------|
         |137 |                 CS VERSION                      |
         |    |-------------------------------------------------|
         \140 |               CCLOSE PLABEL                     |
              |-------------------------------------------------|
          141 |      LOGICAL PROCESS TABLE (PROGEN)            0|
              |-------------------------------------------------|
          142 |                                                 |
              |-------------------------------------------------|
          143 |       LOGICAL PROCESS TABLE (UCOP)             2|
              |-------------------------------------------------|
          144 |       LOGICAL PROCESS TABLE (PFAIL)            3|
              |-------------------------------------------------|
          145 |      LOGICAL PROCESS TABLE (DEVREC)            4|
              |-------------------------------------------------|
          146 |       LOGICAL PROCESS TABLE (DRUSG)            5|
              |-------------------------------------------------|
          147 |       LOGICAL PROCESS TABLE (STMSG)            6|
              |-------------------------------------------------|
          150 |        LOGICAL PROCESS TABLE (LOG)             7|
              |-------------------------------------------------|
          151 |       LOGICAL PROCESS TABLE (LOAD)             8|
              |-------------------------------------------------|
          152 |    LOGICAL PROCESS TABLE (IOMESSPROC)          9|
              |-------------------------------------------------|
          153 |      LOGICAL PROCESS TABLE (SYSIOPRDC)        10|
              |-------------------------------------------------|
          154 |       LOGICAL PROCESS TABLE MEMLOGP           11|
              |-------------------------------------------------|
          155 |      EXTERNAL PLABEL OF "TERMINATE"             |
              |-------------------------------------------------|
          156 |      INTERNAL PLABEL OF "TERMINATE"             |
              |-------------------------------------------------|
```

System Global Area (Cont.)

```
              |-------------------------------------------------|
          157 |     EXTERNAL PLABEL OF "COMMANDINTERP"          |
              |-------------------------------------------------|
          160 |     INTERNAL PLABEL OF "COMMANDINTERP"          |
              |-------------------------------------------------|
          161 |       EXTERNAL PLABLE OF "SPOOLIN"              |
              |-------------------------------------------------|
          162 |       INTERNAL PLABLE OF "TRACEO"               |
              |-------------------------------------------------|
          163 |       EXTERNAL PLABEL OF "TRACEO"               |
              |-------------------------------------------------|
          164 |       INTERNAL PLABEL OF "SPOOLIN"              |
              |-------------------------------------------------|
          165 |       EXTERNAL PLABLE OF "SPOOLOUT"             |
              |-------------------------------------------------|
          166 |       INTERNAL PLABEL OF "SPOOLOUT"             |
   -----      |-------------------------------------------------|
     |    167 |                  3 WORD                         |
     |        |-                                               -|
     |    170 |                  LOGGING                        |
     |        |-                                               -|
     |    171 |                   MASK                          |
     |        |-------------------------------------------------|
     |    172 |STATE|                 DSTN - BUFFER 0           |  STATE:
     |        |-------------------------------------------------|   0 EMPTY
     |    173 |STATE|                 DSTN - BUFFER 1           |   1 CUR
     |        |-------------------------------------------------|   2 FULL
     |    174 |           BUFFER LENGTH (SECTORS)               |
     |        |-------------------------------------------------|
     |    175 |             FREE AREA POINTER                   |
     |        |-------------------------------------------------|
     |    176 |                  FLAGX                          |
 RESERVED     |-------------------------------------------------|
   FOR    177 |       # RECORDS WRITTEN IN BUFFER 0             |
 LOGGING      |-------------------------------------------------|
     |    200 |       # RECORDS WRITTEN IN BUFFER 1             |
     |        |-------------------------------------------------|
     |    201 |       FILE SIZE (BLOCKS) - 1ST HALF             |
     |        |-------------------------------------------------|
     |    202 |       FILE SIZE (BLOCKS) - 2ND HALF             |
     |        |-------------------------------------------------|
     |    203 |             (LOG FILE SIZE)                     |
     |        |-------------------------------------------------|
     |    204 |                (BLOCKS)                         |
     |        |-------------------------------------------------|
     |    205 |             LOG FILE NUMBER   (LOGFILENUM)      |
     |        |-------------------------------------------------|
     |    206 | NUMBER OF LOGGING [BLOCKS WRITTEN (1ST HALF)]   |
     |        |-------------------------------------------------|
     |    207 |   BLOCKS WRITTEN [BLOCKS WRITTEN (2ND HALF)]    |
   -----      |-------------------------------------------------|
```

System Global Area (Cont.)

```
 -----------  |-----------------------------------------|
   |     210|         (TOTAL # LOG RECORDS MISSED)     |
   |        |-----------------------------------------|
   |     211|              (DUE TO LOG FAILURE)        |
   |        |-----------------------------------------|
   |     212| TOTAL# RECORDS MISSED - "JOB INITIATION" LOSS |
   |        |-----------------------------------------|
 LOGGING 213|TOTAL# RECORDS MISSED - "JOB TERMINATION" LOSS |
   |        |-----------------------------------------|
  -----  214|  OPERATOR CONSOLE JOBSESSION $ AT STARTUP |
            |-----------------------------------------|
         215|                                          |
         216|         RESERVED FOR KERNEL USE          |
         217|                                          |
            |-----------------------------------------|
         220| MAPPING FIRMWARE FLAG (NON-ZERO=MPE V/E UCODE)|
            |-----------------------------------------|
         221| BANK AND ADDRESS OF MAPPING DST ( INITIALIZED |
         222| BY DISPATCHER DURING LAUNCHING A PROCESS) |
            |-----------------------------------------|
         223| TOTAL SEGMENT NUMBER OF CURRENT PROCESS  |
            |-----------------------------------------|
         224|        TOTAL FREE PHYSICAL CST ENTRIES   |
            |-----------------------------------------|
         225|        HEAD OF FREE PHYSICAL CST LINK    |
            |-----------------------------------------|
         226|              XLST DST NUMBER             |
            |-----------------------------------------|
         227|                                          |
                            RESERVED
         247|                                          |
            |-----------------------------------------|
         250|            HOLE LIST HEAD (BANK)         | HLHEAD
            |-----------------------------------------|
         251|            HOLE LIST HEAD (ADDRESS)      |
            |-----------------------------------------|
         252|            HOLE LIST TAIL (BANK)         | HLTAIL
            |-----------------------------------------|
         253|            HOLE LIST TAIL (ADDRESS)      |
            |-----------------------------------------|
```

System Global Area (Cont.)

```
 -----      |-----------------------------------------|
 SEGMENT 254|           CURRENT WORD COUNT            | XDSCOUNT
  TRACE     |-----------------------------------------|
   |     255|              BUFFER SIZE                | BUFFSIZE
   |        |-----------------------------------------|
   |     256|             MAG TAPE LDEV               | LDEV
   |        |-----------------------------------------|
 -----   257|      TRACE SEGMENT EXTERNAL LABEL       | TLABEL
            |-----------------------------------------|
         260|                 STMON                   |
            |-----------------------------------------|
         261|             MEASINFOTABPTR              |
            |-----------------------------------------|
         262|    MEASUREMENT STATISTICS CLASS MASK    |GCLASSENABLED
            |-----------------------------------------|
         263|      CLASS 0 STATISTICS BANK NUMBER     |MEASSTATXDSBANK
            |-----------------------------------------|
         264|        CLASS 0 STATISTICS ADDRESS       |MEASSTSTXDSBASE
            |-----------------------------------------|
         265|                                         |
            |              SCAN POINT                 |
         266|                                         |
            |-----------------------------------------|
         267|             MEASFLAGS                   | **
            |-----------------------------------------|
         270|                                         |
 -----      |-----------------------------------------|
   |     271|       INDEX OF PCB AT HEAD OF DISPATCHING Q |SYSDISQHEAD
   |        |-----------------------------------------|
   |     272|       INDEX OF PCB AT TAIL OF DISPATCHING Q |SYSDISPQTAIL
   |        |-----------------------------------------|
   |     273|   DST # OF CDT TABLE (DISC CACHING)     |
   |        |-----------------------------------------|
   |     274|   BANK # OF THE CDT TABLE (DISC CACHING) |
 KERNEL     |-----------------------------------------|
   |     275|   ADDRESS OF CDT TABLE (DISC CACHING)   |
   |        |-----------------------------------------|
   |     276|       HELP LOGICAL DEVICE NUMBER        |
   |        |-----------------------------------------|
   |     277|          CURRENT LOGON DST              |  DSTLOGON
 -----      |-----------------------------------------|
   |     300|              (STOP)                     |
   |     301|              (BITS) (see p. 2-15)       |
   |        |-----------------------------------------|
   |     302|          # PROCESS ENTRIES              |
   |        |-----------------------------------------|
   |     303|                                         |
            |-----------------------------------------|
```

System Global Area (Cont.)

```
            |-----------------------------------------|
   |     304|    DEVREC PIN        |        2         |
   |        |-----------------------------------------|
   |     305|                  %20                    |
   |        |-----------------------------------------|
   |     306|     UCOP PIN         |        0         |
   |        |-----------------------------------------|
   |     307|                  %20                    |
 PROCESS    |-----------------------------------------|
 STOP    310|      LOG PIN         |        1         |
 TABLE      |-----------------------------------------|
   |     311|                  %20                    |
   |        |-----------------------------------------|
   |     312|    IOMESS PIN        |        3         |
   |        |-----------------------------------------|
   |     313|                  %20                    |
   |        |-----------------------------------------|
   |     314|    MEMLOG PIN        |        4         |
   |        |-----------------------------------------|
   |     315|                 %20                     |
   |        |-----------------------------------------|
   |     316|               RESERVED                  |
   |        |-----------------------------------------|
   |     317|               Reserved                  |
  ---       |-----------------------------------------|
  ---    320|    DS GLOBAL DATA SEGMENT DST NUMBER    |
   |        |-----------------------------------------|
   |     321|    RESERVED FOR DS/3000 (SET TO ZERO)   |
   |        |-----------------------------------------|
   |     322|    RESERVED FOR DS/3000 (SET TO ZERO)   |
   |        |-----------------------------------------|
   |     323|         SDS LDEV PLABEL                 |
   |        |-----------------------------------------|
  DS     324|    RESERVED FOR DS/3000 (SET TO ZERO)   |
   |        |-----------------------------------------|
   |     325|    RESERVED FOR DS/3000 (SET TO ZERO)   |
   |        |-----------------------------------------|
   |     326|    RESERVED FOR DS/3000 (SET TO ZERO)   |
   |        |-----------------------------------------|
   |     327|    RESERVED FOR DS/3000 (SET TO ZERO)   |
  ---       |-----------------------------------------|
         330|              DISC STATUS                | LAST
            |-----------------------------------------| DISC
         331|       LDEV           |       DISC       | SIO
            |-----------------------------------------| ERROR
         332|                AONESS                   |
            |-----------------------------------------|
         333|               MAXQUEUE                  |
            |-----------------------------------------| JOBPRI
         334|              DEFAULTQUEUE               |
            |-----------------------------------------|
```

System Global Area (Cont.)

```
            |-----------------------------------------|
         335|              DSCHECK PLABEL             |
            |-----------------------------------------|
         336|              DSOPEN PLABEL              |
            |-----------------------------------------|
         337|              DSCLOSE PLABEL             |
            |-----------------------------------------|
         340|          MANAGEWRITE CONV. PLABEL       |
            |-----------------------------------------|
         341|             CONSDSLINE' PLABEL          |
            |-----------------------------------------|
         342|             CXREMOTE PLABEL             |
            |-----------------------------------------|
         343|             CXDSLINE PLABEL             |
            |-----------------------------------------|
         344|              CXRFA PLABEL               |
            |-----------------------------------------|
         345|             DSIMAGE PLABEL              |
            |-----------------------------------------|
         346| DEFAULT LABEL TYPE   | TAPE LBL AUTO REC FUN |
            |-----------------------------------------|
         347| SYSDB PTR TO TERM INIT CHNL PGM (S30/33 ONLY) |
            |-----------------------------------------|
         350| MP|                                  |SD| SOFTDEATH FLAG
            |-----------------------------------------| MEM PRESSURE
         351|                                         |
            |           LAST CYCLE DURATION           |
         352|                                         |
            |-----------------------------------------|
         353|                                         |
            |            CYCLE THRESHOLD              |
         354|                                         |
            |-----------------------------------------|
         355|     BUG CATCH ENABLE CELL               |
            |-----------------------------------------|
         356|   MONITOR BUFFER     |     TIMESTAMP    | MONBUFTO
            |-----------------------------------------|
         357|   MONITOR BUFFER     |     TIMESTAMP    | MONBUFT1
            |-----------------------------------------|
         360|            DSBREAK PLABEL               |
            |-----------------------------------------|
         361| Bank of last memory word                |LAST MEMORY
            |-----------------------------------------|
         362| Base of last memory word                |ADDRESS
            |-----------------------------------------|
        /363|              PVPROC PIN                 |
        |   |-----------------------------------------|
        |364|          PV RECOGNITION COUNT           |
 PRIVATE<   |-----------------------------------------|
 VOLUMES |365|  VMOUNT FLAGS               |AUTO|ALL|ON|
        |   |-----------------------------------------|
```

System Global Area (Cont.)

```
      |----------------------------------------------|
  |366|                                              |
  |   |----------------------------------------------|
  |367|                                              |
      |----------------------------------------------|
  \370|                                              |
      |----------------------------------------------|
   371| MSG CATALOG LDEV |                           |
      |------------------                            |
   372|    MESSAGE CATALOG DISC ADDRESS              |
      |----------------------------------------------|
   373|                 MSG DST                      |
      |----------------------------------------------|
   374|            CONSMPLINE' PLABEL                |
      |----------------------------------------------|
   375|               CONSMRJE PLABEL                |
      |----------------------------------------------|
   376| SYSTEM LEVEL UDC FLAG (1 = SYS UDC'S EXIST)  |
      |----------------------------------------------|
   377|  SYSDB RELATIVE POINTER TO SYSGLOB EXTENSION |
      |----------------------------------------------|
   400|  CPU NUMBER ( Set by the firmware )          |
      |----------------------------------------------|
   401|          MICROCODE MEMORY LOCATIONS          |
  |   |                                              |
  |   | *NOTE THAT THE LOCATIONS USED DEPEND ON THE  |
  |   |  TYPE OF CPU THAT MPE IS RUNNING AND WHETHER |
  |   | A DUMP, POWERFAIL, OR CNTL B/HALT IS PERFORMED|
   402| ---------------------------------------------|

1401 =  DUMPDEVDRT          1410 = S - BANK          1420 = MEMORY SIZE
  02 =  X                     11 = Z                   21 = SYSTEM HALT #
  03 =  DL                    12 = STATUS              22 = ISR
  04 =  DB - BANK             13 = PB - BANK
  05 =  DB                    14 = PB
  06 =  Q                     15 = P
  07 =  S                     16 = PL
                              17 = CIR
```

SysGlob Extension

%200 words long; Pointer found at SysDB + %377

```
     |-------------------------------------------|
 % 0 |        SWAP QUEUE DELAY (*100MS)          |SWAPQDELAY
     |-------------------------------------------|
   1 |  BANK OF FIRST REGION IN LINKED MEMORY    |FIRST
     |-------------------------------------------|MEMORY
   2 |  BASE OF FIRST REGION IN LINKED MEMORY    |REGION
     |-------------------------------------------|
   3 |     GARBAGE COLLECTION ENABLE FLAG        |GARBCOLLENAB
     |-------------------------------------------|
   4 |   MOVE THRESHOLD (IN PAGES, FOR GARB COLL) |MOVETHRESH
     |-------------------------------------------|
   5 |      MAIN MEMORY PAGE SIZE (IN WORDS)     |
     |-------------------------------------------|
   6 |              VDS PAGE SIZE                |
     |-------------------------------------------|
   7 |                                           |
     |           LAST MAKE ROOM TIME             |
  10 |                                           |
     |-------------------------------------------|
  11 |    MEMORY PRESSURE DURATION THRESHOLD     |
     |-------------------------------------------|
  12 |    RESERVED FOR NATIVE LANGUAGE SUPPORT   |
     |-------------------------------------------|
  13 |    RESERVED FOR NATIVE LANGUAGE SUPPORT   |
     |-------------------------------------------|
  14 |    BAUD RATE OF THE SYSTEM CONSOLE        |
     |-------------------------------------------|
  15 |///////////////////////////////////////////|
     |-------------------------------------------|
  16 |         PLABEL FOR REMOTE'MPE             |
     |-------------------------------------------|

  56
     |-------------------------------------------|
     |-------------------------------------------|
  57 |///////////////////////////////////////////|
     |-------------------------------------------|
  60 |         PLABEL USERLOG (EXTERNAL)         |
     |-------------------------------------------|
  61 |         PLABEL USERLOG (INTERNAL)         |
     |-------------------------------------------|
  62 |         PLABEL RECLOG  (EXTERNAL)         |
     |-------------------------------------------|
```

SysGlob Extension (Cont.)

```
  63|          PLABEL RECLOG  (INTERNAL)             |
    |------------------------------------------------|
  64|          PLABEL RESTART (EXTERNAL)             |
    |------------------------------------------------|
  65|          PLABEL RESTART (INTERNAL)             |
    |------------------------------------------------|
  66|          PMBC LOW CORE BANK #  (USER)          |
    |------------------------------------------------|
  67|          PMBC LOW CORE ADDRESS (USER)          |
    |------------------------------------------------|
  70|          RESERVED FOR IMAGE                    |
    |------------------------------------------------|
  71|        RESERVED FOR MEASIO          12| MIOCNT | *
    |------------------------------------------------|
  72|    LOADER CACHE SEGMENT NUMBER                 |
    |------------------------------------------------|
  73|    PLABEL 3270    (EXTERNAL)                   |
    |------------------------------------------------|
  74|            VERSION                             |
    |------------------------------------------------|
  75|            UPDATE                              |
    |------------------------------------------------|
  76|            FIX                                 |
    |------------------------------------------------|
  77|       COUNT OF TAPE CONTROLLERS USING MEASIO   |
    |------------------------------------------------|
 100|        PORT DATA SEGMENT NUMBER                |
    |------------------------------------------------|
 101|   RESERVED FOR SECOND PORT DATA SEGMENT        |
    |------------------------------------------------|
 102|  SYSTEM FPMAP OPTION FLAG                      |SYSFPMAP
    |------------------------------------------------|
 103|                                                |
 104|                 GLOBAL                         |
 105|                 ALLOW                          |
 106|                 MASK                           |
 107|                                                |
 110|                                                |
    |------------------------------------------------|
 111|                   RESERVED                     |

 117|                                                |
    |------------------------------------------------|
 120| SYS PORT PROCESS PCB RELATIVE INDEX            |
    |------------------------------------------------|
 121|         GLOBAL AFT DST NUMBER                  |
```

SysGlob Extension (Cont.)

```
    |-----------------------------------------------|
 122|     INITIAL/PROGEN COMM. DSEG NUMBER          |
    |-----------------------------------------------|
 123|                                               |

 127|          CURRENTLY UNASSIGNED                 |
    |-----------------------------------------------|
 130| (DS,NETWORK MGMT,APPLICATION SERVICES)        |
    |---                                        ----|
 131|                                               |
    |---                                        ----|
 132|                                               |
    |---                                        ----|
 133|                                               |
    |---                                        ----|
 134|                                               |
    |---                                        ----|
 135|                                               |
    |---                                        ----|
 136|                                               |
    |---                                        ----|
 137|                                               |
    |---                                        ----|
 140|                                               |
    |---                                        ----|
 141|                                               |
    |---                                        ----|
 142|                                               |
    |---                                        ----|
 143|                                               |
    |---                                        ----|
 144|                                               |
    |-----------------------------------------------|
 145|              RESERVED FOR SPL                 |
    |------------                   ----------------|
 146|                 PATH FLOW                     |
    |------------                   ----------------|
 147|                 ANALYZER                      |
    |------------                   ----------------|
 150|                                               |
    |-----------------------------------------------|
 151|            CURRENTLY UNASSIGNED               |
    |------------                   ----------------|

    |------------                   ----------------|
 200|                                               |
    |-----------------------------------------------|

  * MIOCNT = MEASIOCOUNT  (3 BITS)
 ** MEASFLAGS   (15:1) = 1 ==> MONITOR ENABLED
```

```
(14:1) = 1 ==> BUFFER FLIP/FLOP
(13:1) = 1 ==> EOT ON MONITOR TAPE
```

SYSDB Words

System tables may be accessed by using the LST/SST instructions. Pointers have the following format:

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|-----------------------------------------------|
|       Address                 |    Bank       |
|-----------------------------------------------|
```

Address is the whole word with "Bank" masked out to 00000.

Systems that have MPE V/E microcode (all 6X systems, 4X systems with new boards) can have a non-zero bank number. Systems running pre-MPE V/E microcode can only use bank 0, therefore the pointer will look like:

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|-----------------------------------------------|
|      Address                                  |
|-----------------------------------------------|
```

SysGlob Word Definitions

ADDRESS	NAME	FUNCTION
DB+55	BUSY	- SYSDB relative pointer to BUSY TABLE for I/O resources
DB+56	HEAD	- SYSDB relative pointer to table containing head pointers to I/O resource queues
DB+57	TAIL	- SYSDB relative pointer to table containing head pointers to tail of I/O resource queues
DB+60	SIO COUNT	- Number of I/O Programs currently executing
DB+72	POWER FAIL	- 0-no power fail 1-system disc recovery 2-all other disc recovery 3-all other device recovery
DB+73	SYSUP	- System is up and operable
DB+74	CONSLDEVN	- System console logical device number
DB+400	CPU NUMBER	- Set when system aborts

JOBSYNCH job synchronization via jobsynch (sysglob+121(8))

(13:1) - JOBSREADY - set by DEVREC & MORGUE (via procedure STARTDEVICE) indicating a ready job. This prevents UCOP from going to a wait state when a job is just made ready.

(15:1) - DEVFREED - set by DEALLOCATE when device count goes to 0.

NOTE - Both bits above used for synchronization of job-made-ready or devicefreed when UCOP is running.

(14:1) - JOBSWAITING- set by UCOP just before waiting if any job is waiting for list device. Signals DEALLOCATE to awake UCOP when a device is freed.

Allow Mask Format

The Allow mask for MPE V is expanded to six words. There is a mask in each user's JIT and in the SYSGLOB area. The Allow mask contains enough bits for a one-to-one correspondence to every present OPERATOR type command, or any future OPERATOR command. When a user is ALLOWed any OPERATOR command or ASSOCIATED to a device (which will use OPERATOR type commands) then the corresponding bit(s) in the mask in that user's JIT for that command is set. If the ALLOW or ASSOCIATE was done on a global scale, then the bit(s) in the mask of the SYSGLOB area is/are updated.

The following EQUATEs define the mask bit for each operator command.

The first set of commands define the operator commands dealing with devices.

When adding a new command to this set of EQUATEs, be sure to add a corresponding move statement in LOGIMAGE, even if the command will not be logged.

	Word	Bit	#
ABORTIO	0	0	0
ACCEPT	0	1	1
DOWN	0	2	2
GIVE	0	3	3
HEADOFF	0	4	4
HEADON	0	5	5
REFUSE	0	6	6
REPLY	0	7	7
STARTSPOOL	0	8	8
TAKE	0	9	9
UP	0	10	10
MPLINE	0	11	11
DSCONTROL	0	12	12

UPPER LIMIT->DEVICE COMMANDS

ABORTJOB	0	13	13
ALLOW	0	14	14
ALTFILE	0	15	15
ALTJOB	1	0	16
BREAKJOB	1	1	17
DELETE	1	2	18
DISALLOW	1	3	19
JOBFENCE	1	4	20
LIMIT	1	5	21
STOPSPOOL	1	6	22
SUSPENDSPOOL	1	7	23
OUTFENCE	1	8	24
RECALL	1	9	25
RESUMEJOB	1	10	26
RESUMESPOOL	1	11	27
STREAMS	1	12	28
CONSOLE	1	13	29

Allow Mask (Cont.)

	Word	Bit	#
WARN	1	14	30
WELCOME	1	15	31
MON	2	0	32
MOFF	2	1	33
VMOUNT	2	2	34
LMOUNT	2	3	35
LDISMOUNT	2	4	36
MRJECONTROL	2	5	37
JOBSECURITY	2	6	38
DOWNLOAD	2	7	39
MIOENABLE	2	8	40
MIODISABLE	2	9	41
LOG	2	10	42
FOREIGN	2	11	43
IMF	2	12	44
SHOWCOM	2	13	45
OPENQ	2	14	46
SHUTQ	2	15	47
DISCRPS	3	2	48

Logging Related Locations

```
SYSDB
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15

 172 |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 or  |STATE|                        DST #            |
 173 |-----------------------------------------------|

      STATE = 0 if respective buffer empty
              1 if respective buffer is current
              2 if respective buffer is full
```

FLAGX

```
SYSDB
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|---|--|--|
 176 |//////////////////////////////////|SF|HF|BUF|SL|SD|
     |-------------------------------------------------|

 SF = 1 if soft failure
 HF = 1 if hard failure
BUF = 0 if current log buffer is buffer 0
    = 1 if current log buffer is buffer 1
 SL = 1 to indicate a switch in log buffers (from 0 to 1 or from 1 to 0)
 SD = 1 to indicate shutdown in progress
```

Process Stop List General Layout

```
SYSDB
     |---------------------------------------------|
 300 |        STOP BITS REPRESENTING WHICH         |
     |      PROCESSES TO STOP ON "SHUTDOWN"        |
     |---------------------------------------------|
     |             # PROCESS ENTRIES               |
     |---------------------------------------------|
     |/////////////////////////////////////////////|
     |---------------------------------------------|
     |             1ST PROCESS ENTRY               |
     |---------------------------------------------|
     |             2ND PROCESS ENTRY               |
     |---------------------------------------------|
     |                     .                       |
     |                     .                       |
     |                     .                       |
     |                     .                       |
     |---------------------------------------------|
 317 |            LAST PROCESS ENTRY               |
     |---------------------------------------------|
```

Entry Format

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|     PROCESS PIN #      |      STOP BIT #       |
|-----------------------------------------------|
|              PROCESS WAIT STATE               |
|-----------------------------------------------|
```

Preassigned Entries

entry #	process	stop bit #
1	devrec	2
2	ucop	0
3	log	1

Initial Memory Allocation

This section is a description of the method used by INITIAL to allocate memory for MPE tables and code segments in MPE V/E. All memory allocated by INITIAL is permanently allocated. All non-core resident code and data is put on disc before exiting INITIAL.

At the most basic level INITIAL will try to build memory to look exactly as diagrammed below. There are, however, several ways in which to deviate from this structure. Before going into the sources of these deviations, it is necessary to point out which portions of memory are used by INITIAL during the restart and therefore cannot be used by MPE until INITIAL has finished.

Before INITIAL begins to allocate any memory space, it relocates its core resident code, its code segment swapping area and its stack to the highest configured memory space. Additionally, it uses the last %326 words of bank 0 on series 4x machines for its I/O buffer area and temporary code segment table. After INITIAL has built all of core resident MPE (tables and code), it builds the disc resident MPE tables. Since some of the disc resident tables may be too large to be built in INITIAL's stack, these tables are built in unused memory space. Therefore, in addition to the memory space required for INITIAL's code, INITIAL's stack and core resident MPE, there must be enough space left in which to build the largest of the disc resident tables.

For Series 6x machines with the MPE V/E firmware, INITIAL will build the tables with ">" signs by them out of Bank 0 if necessary. For all other tables, INITIAL will essentially build memory in the order shown below. There may be an unused fragment of memory between the DRT's and the system global area which INITIAL will fill with the smaller tables. Neither the tables marked with an asterisk nor the code segments will ever be put in this area. NOTE: INITIAL will build all tables on 32-word boundaries.

If the system being built by INITIAL is configured with 128K words or 160K words of memory then INITIAL's stack will be in bank 1 (the code also on a 128K word memory size). If INITIAL is occupying part of bank 1 and the space is needed for a core resident MPE code segment or to build a disc resident table then INITIAL will print the error message "ERROR #350 OUT OF MEMORY".

Except for the exceptions stated above, for every allocation of memory INITIAL will first try to allocate any remaining space between the DRT's and SYSDB. It will then try the next available space in bank 0, then the next available space in bank 1. If it were necessary it could continue searching until all all banks were checked for available space.

Immediately before exiting INITIAL, INITIAL lays down all the memory region headers and trailers as shown below. For any one bank of memory there will only be one block of core resident MPE, regardless of its contents. The only block of core resident MPE that does not have a reserved region global header is in bank 0. It does have the reserved region global trailer though. Before placing any code outside bank 0 the first 24 words of every bank (except bank 0) is reserved for the region global header.

Bank 0

```
|-------------------------------|
|        Low Core memory        |
|-------------------------------|
|             >DRT              |(Only on 64/68 if Pri-
|-------------------------------|  vilege Mode Bounds
|      System Global area       | Checking is enabled.)
|-------------------------------|
|         Firmware area         |
|-------------------------------|
|       SYSGLOB Extension       |
|-------------------------------|
|         DST/CST/CSTX          |
|-------------------------------|
|              ICS              |
|-------------------------------|
|             PMBC              |(Only for 64/68 if Pri-
|-------------------------------|  vilege Mode Bounds
|            ILT/DIT            | Checking is enabled.)
|-------------------------------|
|              DLT              |
|-------------------------------|
|        Resource Tables        |
|-------------------------------|
|           CST Block           |
|-------------------------------|
|   >Memory Measurement Info    |
|-------------------------------|
|          VDSM Table           |
|-------------------------------|
|       Job Process Count       |
|-------------------------------|
|     > PRI/SEC MSR             |
|-------------------------------|
|             >PCB              |
|-------------------------------|
|     > Swap Table (SLL)        |
|-------------------------------|
|    >Special Request Table     |
|-------------------------------|
|       >Job Cutoff Table       |
|-------------------------------|
|      >Timer Request List      |
|-------------------------------|
|       >System Buffers         |
|-------------------------------|
|             >LPDT             |
|-------------------------------|
|             >IOQ              |
|-------------------------------|
|             >SIR              |
|-------------------------------|
|          >MON Table           |
```

Bank 0 (Cont.)

```
|--------------------------------|
|                                |
| Core Resident CST's in CST order |
|                                |
|                                |
|--------------------------------|
| Reserved Region Global Trailer |
|--------------------------------|
| Available Region Global Header |
|--------------------------------|
|                                |
|                                |
|                                |
|        Available Memory        |
|                                |
|                                |
|                                |
|--------------------------------|
| Available Region Global Trailer |
|--------------------------------|
```

NOTE: The > means these tables can move out of Bank 0 if necessary.

Bank 1

```
|--------------------------------|
| Reserved Region Global Header  |
|--------------------------------|
|                                |
|                                |
|                                |
|    Core Resident CST's and     |
|   tables marked with ">" that  |
|      didn't fit in BANK 0      |
|                                |
|                                |
|                                |
|--------------------------------|
| Reserved Region Global Trailer |
|--------------------------------|
```

CHAPTER 2 MEMORY MANAGEMENT TABLES

Segment Table Structure

The current location and state of each data segment and loaded code segment is maintained in the Segment Table. This table is partitioned into three separate tables as shown in Figure 2-1. The partitions are based on the segment classes: a segment is a data segment, a segment is a system SL segment, or a segment is part of a program. The structure and format of each partition is described in the following.

```
(%2),(%1002)+SYSBASE -------->    +-------+
                                  |       |
                                  |  DST  |
                                  |       |
(%0),(%1001)+SYSBASE -------->    +-------+
                                  |       |
                                  |  CST  |
                                  |       |
                                  +-------+
            CSTXMAP     +-------> | First |
           +-------+    |         |LOADED |
(%1051) -->|       |    |         |PROGRAM|
           +-------+    |         |       |
           |       |----+ +------>+-------+<-- (%3), CURRENT PROGRAM POINTER
           +-------+      |       | NEXT  |
           |       |      |       |LOADED |
           |       |      |       |PROGRAM|
           |       |      |       +-------+
           +-------+      |       |       |
           |       |-------+       |       |
           +-------+               |       |
           |       |               +-------+
           |       |
           +-------+
```

Overall ST Structure

Pointers and DST #'s of Segment Table Components

i. DST

% 2 absolute address of entry 0 of the DST. %1002 sysbase relative index of entry 0 of DST. DST number 2 is the DST Table dst #.

ii. CST

% 0 absolute address of entry 0 of System SL. %1001 sysbase relative index of entry 0 of System SL. %1032 displacement from DST base of entry 0 of System SL (i.e. @CST(last) - @DST(0) = DFS). DST number 4 is the CSTX Table DST #.

iii. CSTX

% 1 absolute address of entry 0 of current program. %1033 displacement from DST base to first CSTX entry SL. DST number 4 is the CSTX Table DST #.

iv. CSTXMAP

%1051 sysbase relative index of entry 0 of CSTXMAP. DST number 43 (%72) is CSTXMAP Table DST #.

Standard Object Identifier Format

```
            0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          |-----------------------------------------------|
          |  TYPE    |              CSTBLK                |
          |-----------------------------------------------|
          |                 OBJECT NUMBER                 |
          |-----------------------------------------------|
```

```
OBJIDENTIFIER(0).(0:4) ==> TYPE
                         = 0 Object is a Data segment
                         = 1 Object is an SL segment
                         = 2 Object is a Program segment
                         = 3 Object is a Cache Domain

OBJIDENTIFIER(0).(4:12) ==> Program index into CSTXBLK
OBJIDENTIFIER(1).(0:16) ==> Number field:
                              DST, CST, CSTX, or CDT number
```

DST Entry Formats

DST/CST Entry 0 Format

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 WORD 0|  # CONFIGURED ENTRIES                         |
        |-----------------------------------------------|
 WORD 1|     ENTRY LENGTH (4)                          |
        |-----------------------------------------------|
 WORD 2|  # AVAILABLE ENTRIES                          |
        |-----------------------------------------------|
 WORD 3|  TABLE RELATIVE INDEX TO FIRST FREE ENTRY     |
        |-----------------------------------------------|
```

DST General Entry Format

Case (i) DST Entry for a Present Data Segment

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
WORD 0   |A |O |R |                SIZE/4                | FIRMINFO
         |-----------------------------------------------|
WORD 1   |D |R |I |S |M |F |S |C |W|                     |
         |C |O |M |T |O |W |Y |O |D|     VMALLOC         |  FLAGS
         |V |C |I |K |D |I |S |R | |                     |
         |  |  |  |  |  |P |  |E | |                     |
         |-----------------------------------------------|
WORD 2   |                   BANK                        | MMBANK
         |-----------------------------------------------|
WORD 3   |                   BASE                        | MMBASE
         |-----------------------------------------------|
```

Case (ii) DST Entry for an Absent Data Segment

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
WORD 0   |A |O |R |                SIZE/4                | FIRMINFO
         |-----------------------------------------------|
WORD 1   |D |R |I |S |M |F |S |C |W|                     |
         |C |O |M |T |O |W |Y |O |D|     VMALLOC         |  FLAGS
         |V |C |I |K |D |I |S |R | |                     |
         |  |  |  |  |  |P |  |E | |                     |
         |-----------------------------------------------|
WORD 2   |     LDEV #             |         HODA         | HODA
         |-----------------------------------------------|
WORD 3   |                   LODA                        | LODA
         |-----------------------------------------------|
```

CST Entry Formats

CST General Entry Format

Case (i) CST Entry for a Present SL Segment or CSTX Segment

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
WORD 0   |A |M |R |T |            SIZE/4                 | FIRMINFO
         |-----------------------------------------------|
WORD 1   |/ |R |I |/ |/ |/ |S |C | ///////////////////// |
         |/ |O |M |/ |/ | /|Y |O |////////////////////// | FLAGS
         |/ |C |I |/ |/ |/ |S |R |////////////////////// |
         |/ |  |  |/ |/ |/ |  |E |////////////////////// |
         |-----------------------------------------------|
WORD 2   |                  BANK                         | MMBANK
         |-----------------------------------------------|
WORD 3   |                  BASE                         | MMBASE
         |-----------------------------------------------|
```

CASE (ii) CST Entry For An Absent Segment SL or CSTX Segment

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
WORD 0   |A |M |R |T |            SIZE/4                 | FIRMINFO
         |-----------------------------------------------|
WORD 1   |/ |R |I |/ |/ |/ |S |C | ///////////////////// |
         |/ |O |M |/ |/ | /|Y |O |////////////////////// | FLAGS
         |/ |C |I |/ |/ |/ |S |R |////////////////////// |
         |/ |  |  |/ |/ |/ |  |E |////////////////////// |
         |-----------------------------------------------|
WORD 2   |    LDEV #             |      HODA             | HODA
         |-----------------------------------------------|
WORD 3   |                  LODA                         | LODA
         |-----------------------------------------------|
```

Case (iii) DST/CST Free Entry

```
|-----------------------------------------|
|                %100000                  |
|-----------------------------------------|
| TABLE RELATIVE OFFSET TO NEXT FREE ENTRY |
|-----------------------------------------|
| TABLE RELATIVE OFFSET TO PREVIOUS FREE ENTRY |
|-----------------------------------------|
| /////////////////////////////////////// |
|-----------------------------------------|
```

Refer to the Logical Segment Table Format in Chapter 11 for more information on XCST.

ST Entry Field Descriptions

```
       A = 1 ==> segment absent
       M = 1 ==> segment privileged
       R = 1 ==> segment has been referenced
       T = 1 ==> segment is being traced
     DCV = 1 ==> disc copy is valid
     STK = 1 ==> segment is a stack
     MOD = 1 ==> a segment modification (exp., contr.) is pending
     FWIP= 1  ==> a forced write of this segment is in progress
     VMPAGECNT = # of virtual memory pages allocated to this segment
     ROC = 1 ==> segment is recoverable overlay candidate
     IMI = 1 ==> segment is in motion in
     SYS = 1 ==> segment is a system segment
     CORE= 1 ==> segment is core resident
     WD= 1   ==> write disabled
```

CSTBLK Format

```
CSTBLK(0)------------------------------------------
        0                                         *
        *   NUMBER OF ENTRIES IN TABLE            *
        -----------------------------------------
        1                                         *
        *   ANY UNASSIGNED ENTRY = -1             *
        -----------------------------------------
        2                                         *
        *   ANY ASSIGNED ENTRY > 0                *
        -----------------------------------------
        3                                         *
        *   REMAINING CSTBLK TABLE ENTRIES        *
        -----------------------------------------
```

The table is initialized to minus one in each entry. When selected, the entry is replaced by a DST-relative index to the entry #0 of the CST extension block. This is the the overhead entry for the associated program.

Program Blocks and the CSTXMAP

Since programs can be dynamically loaded and unloaded, the segment table must be kept packed or fragmentation would occur. Thus, the block of ST entries for a program segment begins at an ST entry number that changes if a program which was loaded before it gets unloaded. To manage this dynamic structure, an auxiliary structure, the CSTXMAP is used. A program is identified by its index, CSTXEIX, into this map. The program's current beginning physical ST entry number is equal to equal to CSTMAP (CSTXEIX).

Entry Format - CST Extension Block

```
CSTXMAP(CSTXEIX)-->----------------------------
                 0  * M = # OF CST'S IN BLOCK  *
                    ---------------------------
                 1  * VALIDITY=%125252         *
                    ---------------------------
                 2  * # OF USERS SHARING BLOCK *
                    ---------------------------
                 3  *           0              *
XCST                ---------------------------     NON-XCST
  1 ------------>   * HAS CST ENTRY FORMAT     *  <---- %301
                    ---------------------------
  2 ------------>   * HAS CST ENTRY FORMAT     *  <---- %302
                    ---------------------------
                                .
                                .
                    ---------------------------
  +M ---------->    * HAS CST ENTRY FORMAT     *  <--- %303
                    ---------------------------
```

The value of CSTXEIX is established when a CST extension block is allocated. This index into the array CSTXMAP is maintained in the PCB of each process sharing the block.

Fixed DST Entry Assignments

OCTAL		DECIMAL	TABLE NAME
0		0	
1	CST	1	CST
2	DST	2	DST
3	PCB	3	PCB
4	CSTX	4	CSTX
5	SYSTEM GLOBAL AREA	5	SYS
6	CORE	6	CORE
7	ICS	7	ICS
10	SYSTEM BUFFERS	8	SBUF
11	UCOP REQUEST QUEUE	9	UCRQ
12	PROCESS-PROCESS COMMUNICATION TABLE	10	PPCOM
13	I/O QUEUE	11	IOQ
14	TERMINAL BUFFERS	12	TBUF
15	LOGICAL-PHYSICAL DEVICE TABLE	13	LPDT
16	LOGICAL DEVICE TABLE	14	LDT
17	DRIVER LINKAGE TABLE	15	DLT
20	I/O RESOURCE TABLES	16	BUSY, HEAD, TAIL
21	SECONDARY MSG TABLE	17	SECMSGTAB
22	LOADER SEGMENT TABLE	18	LST
23	TIMER REQUEST LIST	19	TRL
24	DIRECTORY	20	DDS

DST (Cont.)

OCTAL		DECIMAL	TABLE NAME
25	DIRECTORY SPACE	21	
26	RIN TABLE	22	RIN
27	SWAPTABLE (SLL)	23	SWAPTAB
30	JOB PROCESS COUNT	24	JPCNT
31	JOB MASTER TABLE	25	JMAT
32	TAPE LABEL TABLE	26	VDD
33	LOG TABLE	27	LOGTAB
34	REPLY INFORMATION TABLE	28	RIT
35	VOLUME TABLE	29	VTAB
36	BREAKPOINT TABLE	30	STOP
37	LOG BUFFER1	31	
40	LOG BUFFER2	32	
41	LOG ID TABLE	33	LIDTAB
42	ASSOCIATE TABLE	34	
43	CST BLOCK	35	CSTBLK
44	JOB CUTOFF TABLE	36	JCUT
45	SYSTEM JIT	37	SJIT
46	SPECIAL REQ TABLE	38	SRT
47	VIRTUAL DISC SPACE MANAGEMENT TABLE	39	VDSMTAB
50	DEVICE CLASS TABLE	40	DEVCLASS
51	Reserved Kernel	41	

DST (Cont.)

OCTAL		DECIMAL	TABLE NAME
52	ILT	42	ILT
53	SIR TABLE	43	SIR
54	FMAVT	44	FMAVT
55	INPUT DEVICE DIRECT	45	IDD
56	OUTPUT DEVICE DIRECT	46	ODD
57	WELCOME MESSAGE #1	47	LOGONDSTN1
60	WELCOME MESSAGE #2	48	LOGONDSTN2
61	CS DATA SEGMENT	49	CSTAB
62	PROCESS-JOB CROSS REFERENCE	50	PJXREF
63	SYSTEM JDT	51	SYSJDT
64	COMMAND LOGON DST	52	CILOGDST
65	MOUNTED VOL. SET TABLE	53	MVTAB
66	PRI.VOL. USER TABLE	54	PVUSER
67	RESERVED KERNEL	55	
70	DISC REQUEST TABLE	56	DISCREQTAB
71	MSG HARBOR TABLE	57	MSGHARBTAB
72	PRIMARY MESSAGE TABLE	58	PRIMMSGTAB
73	MEASUREMENT INFO TABLE	59	MEASINFOTAB
74	FIRST FREE DST	60	

Swap Tables

The SWAPTAB is a core resident memory management table used to keep track of the locality lists of the competing processes. The PCB entry for a process has a SWAPTAB relative pointer to the header entry for the process.

SWAPTAB DST# = 23 (%27)

%1004 System table pointer to SWAPTAB entry 0.

NOTE: The number of entries configured will be 3 greater than the number configured via SYSDUMP. (Entry 0 consumes 3 entries).

SWAPTAB Entry 0 Format

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0  |              # ENTRIES CONFIGURED             |  0
     |-----------------------------------------------|
  1  |                 ENTRY SIZE (6)                |  1
     |-----------------------------------------------|
  2  |              # AVAILABLE ENTRIES              |  2
     |-----------------------------------------------|
  3  |    TABLE RELATIVE INDEX OF FIRST FREE ENTRY   |  3
     |-----------------------------------------------|
  4  |    TABLE RELATIVE INDEX OF LAST FREE ENTRY    |  4
     |-----------------------------------------------|
  5  |                HIGH WATER MARK                |  5
     |-----------------------------------------------|
  6  |             # PRIMARY ENTRIES  (0)            |  6
     |-----------------------------------------------|
  7  |     HEAD OF IMPEDED QUEUE (PCB RELATIVE)    |  7
     |---------------------------------------------|
  8  |     TAIL OF IMPEDED QUEUE (PCB RELATIVE)    |  10
     |---------------------------------------------|
  9  |        # CURRENTLY IMPEDED PROCESSES        |  11
     |---------------------------------------------|
 10  |           MAX # OF IMPEDED PROCESSES        |  12
     |---------------------------------------------|
 11  |       CUMULATIVE # OF IMPEDED PROCESSES     |  13
     |-----------------------------------------------|
 12  |                       .                       |  14
     |                       .                       |
     |                       .                       |
     |                                               |
 17  |                                               |  21
     |-----------------------------------------------|
```

SWAPTAB Unassigned Entry Format

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0  |                   %100000                     |
     |-----------------------------------------------|
  1  |     TABLE RELATIVE INDEX OF NEXT FREE ENTRY   |
     |-----------------------------------------------|
  2  |     TABLE RELATIVE INDEX OF PREV. FREE ENTRY  |
     |-----------------------------------------------|
  3  |                       0                       |
     |-----------------------------------------------|
  4  |                       0                       |
     |-----------------------------------------------|
  5  |                       0                       |
     |-----------------------------------------------|
```

An assigned entry in the swaptab is a process' SLL header or a member of a process' SLL. These formats are now described.

Segment Locality Lists (SLL)

The system maintains for each process a segment locality list (SLL) of the segments belonging to that process' current working set. The process' SLL consists of a header and a list of entries. The header and list entries are taken from the SWAPTAB.

A process' SLL is located via the process' PCB entry. PCB01 contains the SLL relative index of the process' SLL header.

```
                     SWAPTAB

             |--------------------------------|
             |               .                |
             |               .                |
             |               .                |
             |--------------------------------|
PCB01-->     |           SLLHEADER            |
          +--|                                |
          |  |--------------------------------|
          |  |               .                |
          |  |               .                |
          |  |               .                |
          |  |--------------------------------|
          +->|         FIRST SLL ENTRY        |
          +--|                                |
          |  |--------------------------------|
          |  |               .                |
          |  |               .                |
          |  |               .                |
          |  |--------------------------------|
          +->|          NEXT SLL ENTRY        |
          +--|                                |
          |  |--------------------------------|
          |  |               .                |
          |  |               .                |
          v  |               .                |
             |--------------------------------|
```

SLL Header Format

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
      |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
      |  |S |H |I |P |S |S |  |                       |
      |  |W |A |N |A |T |W |  |                       |
  0   |  |R |S |T |R |R |I |  |  IOCNT                | SCHEDTOIOMSG
      |  |E |M |L |T |T |P |  |                       |
      |  |Q |E |O |I |O |  |  |                       |
      |  |  |M |C |N |V |  |  |                       |
      |-----------------------------------------------|
  1   |  TABLE RELATIVE INDEX OF FIRST ENTRY IN LIST  | FIRSTINX
      |-----------------------------------------------|
  2   |///////////////////////////////////////////////|
      |-----------------------------------------------|
  3   |  TABLE RELATIVE INDEX OF MEMORY REQUEST ENTRY | MEMREQINX
      |-----------------------------------------------|
  4   |          # ENTRIES IN PROCESS' SLL            | SEGCOUNT
      |-----------------------------------------------|
  5   |///////////////////////////////////////////////|
      |-----------------------------------------------|
```

```
SLL(SLLHEADINX+0)
              .(1:1) SWREQ, Swap Required Flag
              .(2:1) HASMEM, Has Memory Flag
              .(3:1) INTLOC, Initialize locality to minimum
              .(4:1) PARTIN, Process partially swapped in
              .(5:1) STRTOV, Start swap over flag
              .(6:1) SWIP, Swap In Progress Flag
              .(8:8) IOCNT, Segment read completions until awake
```

SLL List Entry Format

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
      |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0   | PCB RELATIVE INDEX OF THE NEXT IMPEDED PIN    | NEXTIMPPIN
      |-----------------------------------------------|
  1   |   TABLE RELATIVE INDEX TO NEXT ENTRY IN LIST  | NEXTINX
      |-----------------------------------------------|
  2   |  TABLE RELATIVE INDEX TO PREV. ENTRY IN LIST  | PREVINX
      |-----------------------------------------------|
  3   |                                               | SLL'OBJDESC
      |-            OBJECT IDENTIFIER                -|
  4   |                                               | SLL'OBJNUM
      |-----------------------------------------------|
  5   |M |S |D |L |B |F |S |T |F |L |D |              | SLL'FLAGS
      |A |T |I |O |L |R |L |O |Z |K |E |   PRE        |
      |P |K |S |C |K |O |L |S |R |R |C |   FETCH      |
      |S |  |C |K |R |Z |I |S |E |E |C |   COUNT      |
      |E |  |I |E |E |E |M |  |Q |Q |N |              |
      |G |  |O |D |Q |N |I |  |  |  |T |              |
      |-----------------------------------------------|
```

```
SLL(SLLINX+0)  NEXTIMPPIN, next make present deferred queue
               PCB Index

SLL(SLLINX+1)  NEXTINX, next SLL entry

SLL(SLLINX+2)  PREVINX, previous SLL entry

SLL(SLLINX+3)  SLL'OBJDESC, 1st word of object identifier

SLL(SLLINX+4)  SLL'OBJNUM, 2nd word of object identifier

SLL(SLLINX+5)
             .(0:1)  MAPSEG, process' CST mapping segment (LSTT)
             .(1:1)  STK, process' stack entry
             .(2:1)  DISCIOSEG, disc I/O pending on this segment
             .(3:1)  LOCKED, segment locked in memory
             .(4:1)  BLKLK, request for blocked lock
             .(5:1)  FROZE, segment frozen in memory
             .(6:1)  SLLIMI, process queued for this segment
             .(7:1)  TOSS, Toss this entry
             .(8:1)  FRZREQ, request segment to be frozen
             .(9:1)  LKREQ, request to lock segment in memory
            .(10:1)  DECCNTFLAG,
            .(11:5)  PREFETCHCOUNT,
```

NOTE:
The Swap Table will be configured with at least twice the number of configured PCBs.

Special Request Table

Used for passing data segment size change info and for keeping a list of devices waiting for a segment to arrive in memory.

%1042 - SRT relative index to entry # 0
%1043 - SRT relative index to the head of the queue

NOTE: The number of entries configured will be 3 greater than the number configured via SYSDUMP. (Entry #0 consumes 3 entries).

SRT Entry 0 Format

```
        -----------------------------------
   0   |   # ENTRIES CONFIGURED            |
       |-----------------------------------|
   1   |      ENTRY SIZE (6)               |
       |-----------------------------------|
   2   |      # AVAILABLE ENTRIES          |
       |-----------------------------------|
   3   | TABLE REL. INDEX OF 1ST FREE ENTRY|
       |-----------------------------------|
   4   |TABLE REL. INDEX OF LAST FREE ENTRY|
       |-----------------------------------|
   5   |         HIGH WATER MARK           |
       |-----------------------------------|
   6   |       # PRIMARY ENTRIES           |
       |-----------------------------------|
   7   | HEAD OF IMPEDED QUEUE (PCB REL.)  |
       |-----------------------------------|
   8   | TAIL OF IMPEDED QUEUE (PCB REL.)  |
       |-----------------------------------|
   9   | # CURRENTLY IMPEDED PROCESSES     |
       |-----------------------------------|
  10   | # MAXIMUM IMPEDED PROCESSES       |
       |-----------------------------------|
  11   | CUMULATIVE # OF IMPEDED PROCESSES |
       |-----------------------------------|
  12   |                                   |
       |                .                  |
       |                .                  |
       |                .                  |
       |                                   |
  17   |                                   |
       |-----------------------------------|
```

The following entry format is for data segment size changes:

```
      |-------------------------------------|
 0    |  NEXT ENTRY FOR DATA SEGMENTS       |
      |-------------------------------------|
 1    |                                     |
      |-       OBJECT IDENTIFIER           -|
 2    |                                     |
      |-------------------------------------|
 3    |       NEW DATA SEGMENT SIZE         |
      |-------------------------------------|
 4    |     READ DISPLACEMENT               |
      |-------------------------------------|
 5    |         MOVE COUNT                  |
      |-------------------------------------|
```

The following is the format for devices waiting on a segment: (The region header for the segment contains an SRT relative index to this entry. If more that 5 devices are waiting on this segment, another entry will be linked to this entry.)

```
      |-------------------------------------|
 0    | NEXT ENTRY OF QUEUED DEVS ON SEG    |
      |-------------------------------------|
 1    |              IOQINX                 |
      |-------------------------------------|
 2    |              IOQINX                 |
      |-------------------------------------|
 3    |              IOQINX                 |
      |-------------------------------------|
 4    |              IOQINX                 |
      |-------------------------------------|
 5    |              IOQINX                 |
      |-------------------------------------|
```

NOTE:
The number of primary configured entries will be equal to the total number of LDEVs configured. The number of secondary entries will be configured to be at least the same as the number of PCBs configured. Data segment change entries are secondary type, while devices queued entries will be primary entries.

Main Memory Region Headers and Trailers

Main memory is partitioned into regions. Each region is in one of three states: available, reserved, or assigned.

An available region is available for consumption by the free space allocation mechanism. An available region consists of neighboring subregions, each of which is either a hole or an overlay candidate. An available region is linked into the available region list.

A reserved region is a main memory region which is in the transition state from available to assigned. A reserved region has been cleaned, and there is a pending disc read of a segment into the region.

Assigned regions are occupied by present segments. Available and reserved regions consist of one or more adjacent subregions. Region headers and trailers are partitioned into global and local components. The global region header/trailer is only valid for the first/last subregion in regions consisting of more than one subregion.

The region headers and trailers of available, reserved, and assigned regions contain the state and control information pertaining to the current or planned contents of the region.

Cache domains are another form of assigned regions and are designated as such in the subregion header. If the cache domain is "mapped" - has I/O pending against it - then the object identifier will have a non-zero value in the second word of the segment identifier field. If the second word of the segment identifier field is zero, then this region is a cache domain that is unmapped. (Refer to Chapter 23 for further information regarding Disc Caching.)

Header length = 24
Trailer length = 4

Global Region Trailer

```
        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
       ------------------------------------------------
RB-27 |      PREVIOUS TRAILER SUBREGION SIZE          | PTSS
      |-----------------------------------------------|
RB-26 |      PREVIOUS TRAILER REGION STATE            | PTRAS
      |A  |R  |A  |///////////////////////////////////|
      |S  |E  |V  |///////////////////////////////////|
      |S  |S  |   |///////////////////////////////////|
      |-----------------------------------------------|
RB-25 |      PREVIOUS TRAILER REGION SIZE             | PTRS
       ------------------------------------------------
```

Global Region Header (Available Regions)

```
        0  1  2  3  4  5  6 *7  8  9 10 11 12 13 14 15
       ------------------------------------------------
RB-24 |       REGION ASSIGNMENT STATE                 | RAS
      |A  |R  |A  |C  |S  |L  |F  |I  |///////////////|M  |
      |S  |E  |V  |L  |C  |K  |Z  |O  |///////////////|I  |
      |S  |S  |   |N  |   |P  |N  |F  |///////////////|P  |
      |   |   |   |D  |   |   |   |Z  |///////////////|   |
      |   |   |   |   |   |   |   |N  |///////////////|   |
      |-----------------------------------------------|
RB-23 |       REGION SIZE                             | RS
      |-----------------------------------------------|
RB-22 |///////////////////////////////////////////////|
       ------------------------------------------------
RB-21 |///////////////////////////////////////////////|
      |-----------------------------------------------|
RB-20 |    PREVIOUS LINK  (ADDRESS OF PL FIELD        | PL
      |-                                             -|
      |       OF PREVIOUS AVAILABLE REGION)           |
      |-----------------------------------------------|
RB-18 |     NEXT LINK  (ADDRESS OF NL FIELD           | NL
      |-                                             -|
      |        IN NEXT AVAILABLE REGION)              |
      |-----------------------------------------------|
RB-16 |///////////////////////////////////////////////|
       ------------------------------------------------
```

Subregion Header (Available Regions)

```
        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
       ------------------------------------------------
RB-15 |      SUBREGION ASSIGNMENT STATE               | SAS
      |C  |R  |R  |/////////////////////////////| I   |
      |A  |E  |O  |/////////////////////////////| O   |
      |C  |F  |C  |/////////////////////////////| S   |
      |H  |   |   |/////////////////////////////| T   |
      |-----------------------------------------------|
RB-14 |              SUBREGION SIZE                   | SS
      |-----------------------------------------------|
RB-13 |V | SUBREGION DISPLACEMENT IN MAIN MEM. PAGES  | SD
      |-----------------------------------------------|
RB-12 |           WRITE REQUEST POINTER               | WREQP
      |-----------------------------------------------|
RB-11 |                                               | OBJIDENT
      |-           OBJECT IDENTIFIER                 -|
      |                                               |
      |-----------------------------------------------|
RB-9  |///////////////////////////////////////////////|
      |-----------------------------------------------|
RB-8  |///////////////////////////////////////////////|
      |-----------------------------------------------|
RB-7  |    LDEV             |         HODA            | HODA
      |-----------------------------------------------|
RB-6  |      Low Order Disk Address                   | LODA
       ------------------------------------------------
RB-5  |///////////////////////////////////////////////|
      |-----------------------------------------------|
RB-4  |///////////////////////////////////////////////|
      |-----------------------------------------------|
RB-3  |///////////////////////////////////////////////|
      |-----------------------------------------------|
RB-2  |///////////////////////////////////////////////|
      |-----------------------------------------------|
RB-1  |///////////////////////////////////////////////|
       ------------------------------------------------
```

Global Region Header (Reserved Regions)

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        -------------------------------------------------
RB-24   |      REGION ASSIGNMENT STATE                  | RAS
        |A |R |A |C |S |L |F |I |////////////////////|M |
        |S |E |V |L |C |K |Z |O |////////////////////|I |
        |S |S |  |N |  |P |N |F |////////////////////|P |
        |  |  |  |D |  |  |  |Z |////////////////////|  |
        |  |  |  |  |  |  |  |N |////////////////////|  |
        |-----------------------------------------------|
RB-23   |      REGION SIZE                              | RS
        |-----------------------------------------------|
RB-22   |     ON GOING I/O COUNT                        | IOCNT
        |-----------------------------------------------|
RB-21   |      INITIATION MESSAGE                       | INITMSG
        |M |E |O |Q |I |E |G |M |R |M |//////////////|M |
        |S |X |N |U |N |X |A |S |E |S |//////////////|S |
        |G |T |G |E |C |P |R |G |L |G |//////////////|G |
        |P |D |O |S |O |R |B |A |P |S |//////////////|V |
        |R |I |I |E |R |E |A |B |A |T |//////////////|A |
        |O |S |N |G |M |Q |G |O |G |A |//////////////|L |
        |C |A |G |R |S |U |E |R |E |R |//////////////|I |
        |E |B |D |E |V |E |  |T |  |T |//////////////|D |
        |S |L |  |  |E |  |  |  |  |  |//////////////|  |
        |-----------------------------------------------|
RB-20   | LOCATION OF DISC REQUEST OR MOVE MSG          | INITINFO
        |-----------------------------------------------|
RB-19   |         COMPLETION MESSAGE                    | COMPMSG
        |M |M |B |S |I |M |//////////////////////////|M |
        |S |O |L |C |O |S |//////////////////////////|S |
        |G |V |K |H |W |G |//////////////////////////|G |
        |P |E |D |E |A |A |//////////////////////////|V |
        |R |R |L |D |I |B |//////////////////////////|A |
        |O |E |K |M |T |O |//////////////////////////|L |
        |C |Q |  |S |  |R |//////////////////////////|I |
        |  |  |  |G |  |T |//////////////////////////|D |
        |-----------------------------------------------|
RB-18   |  MAKE PRESENT DEFERRED QUEUE (PCB INDEX)      | MPQLINK
        |-----------------------------------------------|
RB-17   |      RELEASE PAGE COUNT                       | PAGECNT
        |-----------------------------------------------|
RB-16   | SPECIAL REQUEST TABLE PTR (SRT TABLE REL)     | SPECREQTABPTR
        -------------------------------------------------
```

Subregion Header (Reserved Regions)

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        -------------------------------------------------
RB-15   |     SUBREGION ASSIGNMENT STATE                | SAS
        |C |R |R |/////////////////////////////| I      |
        |A |E |O |/////////////////////////////| O      |
        |C |F |C |/////////////////////////////| S      |
        |H |  |  |/////////////////////////////| T      |
        -------------------------------------------------
RB-14   |               SUBREGION SIZE                  | SS
        |-----------------------------------------------|
RB-13   |V | SUBREGION DISPLACEMENT IN MAIN MEM. PAGES  | SD
        |-----------------------------------------------|
RB-12   |            WRITE REQUEST POINTER              | WREQP
        |-----------------------------------------------|
RB-11   |                                               | OBJIDENT
        |-           OBJECT IDENTIFIER                 -|
        |                                               |
        |-----------------------------------------------|
RB-9    | FREEZE COUNT         | LOCK COUNT             | LKFZCNT
        |-----------------------------------------------|
RB-8    | WRITE DISABLE COUNT  | I/O FROZEN COUNT       | WDIOFZCNT
        |-----------------------------------------------|
RB-7    |   LDEV               | HIGH ORDER DISC ADDRESS| HODA
        |-----------------------------------------------|
RB-6    |           LOW ORDER DISC ADDRESS              | LODA
        |-----------------------------------------------|
RB-5    |///////////////////////////////////////////////|
        |-----------------------------------------------|
RB-4    |///////////////////////////////////////////////|
        |-----------------------------------------------|
RB-3    |                  TIME OF                      | ARRTIME
        |-                                             -|
        |                  ARRIVAL                      |
        |-----------------------------------------------|
RB-1    |///////////////////////////////////////////////|
        |-----------------------------------------------|
```

Subregion Header (Cached Regions)

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        -------------------------------------------------
RB-15   |     SUBREGION ASSIGNMENT STATE                | SAS
        |C |R |R |/////////////////////////////| I      |
        |A |E |O |/////////////////////////////| O      |
        |C |F |C |/////////////////////////////| S      |
        |H |  |  |/////////////////////////////| T      |
        |-----------------------------------------------|
RB-14   |               SUBREGION SIZE                  | SS
        |-----------------------------------------------|
RB-13   |V | SUBREGION DISPLACEMENT IN MAIN MEM. PAGES  | SD
        |-----------------------------------------------|
RB-12   |            WRITE REQUEST POINTER              | WREQP
        |-----------------------------------------------|
RB-11   |                                               | OBJIDENT
        |-           OBJECT IDENTIFIER                 -|
        |                                               |
        |-----------------------------------------------|
RB-9    |   PREVIOUS CACHED REGION (ADDRESS OF PD       | PD
        |-                                             -|
        |      FIELD OF PREVIOUS CACHED REGION)         |
        |-----------------------------------------------|
RB-7    |     LDEV              | HIGH ORDER DISC ADDRESS | HODA
        |-----------------------------------------------|
RB-6    |      LOW ORDER DISC ADDRESS                   | LODA
        |-----------------------------------------------|
RB-5    |   NEXT CACHED REGION (ADDRESS OF ND           | ND
        |-                                             -|
        |      FIELD OF NEXT CACHED REGION)             |
        |-----------------------------------------------|
RB-3    |                TIME OF                        | ARRTIME
        |-                                             -|
        |                ARRIVAL                        |
        |-----------------------------------------------|
RB-1    |    DISC ADDRESS  CSL(8)                       | CACDADISP
        -------------------------------------------------
```

Region Header and Trailer Field Descriptions

RAS,   Region Assignment State
- .(0:1) Region Assigned Flag
- .(1:1) Region Reserved Flag
- .(2:1) Region Available Flag
- .(3:1) Region Cleaned Flag
- .(4:1) Size Change Pending Flag
- .(5:1) Region Locked Flag
- .(6:1) Region Frozen Flag
- .(7:1) Region I/O Frozen Flag
- .(8:1) LSTT segment
- .(9:6) Not used
- .(15:1) Blocked Lock Migration in Progress Flag

IOCNT,   On-Going I/O Count
= # of on-going I/O's in the region which must complete
before the initiation message can be processed.

INITMSG,   Initiation Message
- .(0:1) Message Processed Toggle Switch
- .(1:1) Message Externally Disabled Flag
- .(2:1) Message On-going I/O Disabled Flag
- .(3:1) Queue Segment Read Disc Request Flag
- .(4:1) Incore Move Request Flag
- .(5:1) Expansion Request Flag
- .(6:1) Garbage Collection Flag
- .(7:1) Message Aborted Flag
- .(8:1) Release Residual Pages Flag
- .(9:1) Ok to start completion flag
- .(10:5) Not used
- .(15:1) Message Valid Flag

INITINFO,   Initiation Message Auxiliary Information
= DRQ relative index of segment read disc request if INITMSG.
QREADREQ=1
   or
= +/- Displacement to initiation message for moves
and expansions.

COMPMSG,   Completion Message

- .(0:1) Message Processed Toggle Switch
- .(1:1) Segment Modification Required
- .(2:1) Block Lock Request
- .(3:1) Send Scheduler A Message
- .(4:1) Awaken A Device
- .(5:1) Message Aborted
- .(6:9) Available
- .(15:1) Message Valid Flag

MPQLINK PCB relative index of the HEAD of the make present queue.

PAGECNT, Release Page Count
=# of extra pages to release before processing initiation message.

SPECREQTABPTR, A Special Request Table relative index to the list of devices queued on this segment.

SAS, Subregion Assignment State
.(0:1) Cached region
.(1:1) Referenced
.(2:1) Recover Overlay Candidate
.(13:3) I/O Status from region fetch

SS, Subregion Size

SD, Subregion Displacement
.(0:1) Displacement Count Valid Flag
.(1:15) # Pages to Base of Region

WREQP, Write Request Pointer
= DRQ Relative Index of Disc Write Request when the Data Segment in the Subregion is in Motion Out
When the region belongs to a cashed domain which is mapped (i. e. OBJIDENT = 30000/non zero number) this word is non zero. If the cashed domain is not mapped WREQP is zero.

OBJIDENT, Object Identifier- has standard object identifier format

LKFZCNT, Lock and freeze count
.(0:8) Number of times region has been frozen
.(8:8) Number of times region has been locked

WDIOFZCNT, Iofreeze count
.(0:8) Not used
.(8:8) Number of times region has been iofrozen

For regions belonging to cashed domains, the above two words contain the absolute address of the PD field in the previous region belonging to a cashed domain.

HODA, High order disc address in virtual memory of this region

LODA, Low order disc address in virtual memory of this region

ND, Next cashed domain link for cashed domain regions only. Contains the absolute address of the ND field of the next cashed region.(2 words)

ARRTIME, Arrival time, contains the time at which the segment contained in the region became present

CACDADISP Valid only for regions containing a cashed domain, this word represents the disc address (in one word) of the segment contained in the region. This word which exists in each member of a linked list of cashed domains, is used as the target word during the LLSH instruction.

Space Allocation Structures

As of MPE V/P and V/E, one doubly linked list structure is used instead of the multiple lists ordered by size as in MPE IV. Sysglob locations %250 through %253 contain the respective head and tail (bank & address) of the available region list. These four words have in essence replaced the ARSBM and ARL data structures in MPE IV. Memory allocation and deallocation is handled through PUTONARL and TAKEOFFARL. The search for an available region of the desired size is done via the LLSH instruction. The format of the list is the following :

> Sysglob %250 & %251 points to the absolute address of the NEXT LINK field (two words) in the first available region on the list. The NEXT LINK field in the first available region points to the absolute address of the NEXT LINK field in the second available region and so on. It is worth mentioning that in addition to having a NEXT LINK field, each available region also contains a PREVIOUS LINK pointer, which makes management of the list both easier and faster.

CHAPTER 3 DISC LAYOUT

System Disc Layout

```
SECTOR #                                           SECTOR #
           ----------------------------------------
%  0|              DISC LABEL                   |0
    |---------------------------------------------|
   1|      DEFECTIVE TRACKS/SECTOR TABLE          |1
    |---------------------------------------------|
   2| COLD LOAD CHANNEL PROGRAM FOR  HP-IB        |2
    |---------------------------------------------|
   3| MEM DUMP CHANNEL PROGRAM FOR  HP-IB         |3
    |---------------------------------------------|
   4|                                             |4  \
    |---------------------------------------------|   |
   5|                                             |5  |
    |--------------             ------------------|   |
   6|                                             |6  |
    |-----------     CODE FOR     ----------------|   |
   7|             INITIAL PROGRAMS                |   |
    |-----------     "BOOTSTRAP"   ---------------|   |
  10|                 SEGMENT                     |   |
    |-------------             -------------------|   |
  11|                                             |   |
    |---------------            ------------------|   > VARIABLE
   .|                                             |   | LENGTH
    |---------------------------------------------|   |
   .|                                             |   |
    |---------------------------------------------|   |
   .|                                             |   |
    |---------------------------------------------|   |
   .|                                             |   |
    |---------------------------------------------|   |
    |                                             |   |
    |---------------------------------------------|   |
    |                                             |   /
    |---------------------------------------------|
    | LOW CORE (CST POINTER, QI, ZI, POINTER)     |  <--| FOLLOWS
    |---------------------------------------------|     | IMMEDIATELY
    |   TEMPORARY CST (INITIAL PROGRAM)           |     | AFTER
    |---------------------------------------------|     | BOOTSTRAP
    |       INTERNAL INTERRUPT HALTS              |     | SEGMENT
    |---------------------------------------------|
    |            BOOTSTRAP STACK                  |
    |---------------------------------------------|
    |   REMAINDER OF SIO COLD LOAD PROGRAM        |
    |---------------------------------------------|
```

System Disc Layout (Cont.)

```
SECTOR # |---------------------------------------------|SECTOR #
  %      |                                             |
         |---------------------------------------------|
         |                                             |
         |---------------------------------------------| .
        .|                                             | .
         |---------------------------------------------| .
       34|    DISC COLD LOAD INFORMATION TABLE         |28
         |---------------------------------------------|
       35|    DISC COLD LOAD INFORMATION TABLE         |29
         |---------------------------------------------|
       36|    DISC COLD LOAD INFORMATION TABLE         |30
         |---------------------------------------------|
       37|    SYSDUMP/INITIAL COMMUNICATION RECORD     |31
         |---------------------------------------------|
       40|    DISC COLD LOAD INFO. TABLE EXT.          |32
         |---------------------------------------------|
       41|    DISC COLD LOAD INFO. TABLE EXT.          |33
         |---------------------------------------------|
         |                                             |
```

System Disc Layout (Cont.)

```
SYSDB    |---------------------------------------------|
----->   |                                             | ---> NOTE: INITIAL
%130/131 |                                             |      TRIES TO
         |          SYSTEM DIRECTORY                   |      ALLOCATE
         |                                             |      DIRECTLY AFTER
         |---------------------------------------------|      THE FREE SPACE
         |                                             |      MAP.  HOWEVER,
         |                                             |      THIS MAY
         |                                             |      VARY DEPENDING
         |          VIRTUAL MEMORY AREA                |      ON DELETED
         |                                             |      OR REASSIGNED
         |                                             |      TRACKS
         |                                             |
         |---------------------------------------------|
         |        INITIAL PROGRAM SEGMENTS             |
         |        (EXCEPT BOOTSTRAP SEG)               |
         |---------------------------------------------|
         |             SYSTEM FILES                    |
         |        (FROM COLD LOAD TAPE)                |
         |---------------------------------------------|
         |                                             |
         |       VOLUME TABLE                          |
         |       INITIAL PROGRAM STACK                 |
         |       REMAINING INITIAL CODE SEGMENTS       |
         |                                             |
         |---------------------------------------------|
         |                                             |
         |             USER FILES                      |
         |                 .                           |
         |                 .                           |
         |                 .                           |
         |                                             |
         |                                             |
         |                                             |
         |                                             |
          ---------------------------------------------
```

Disc Label (Sector 0 of Disc)

System Volume

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0|                CONTROL ORDER                   |0
   |                                                |
  1|                <<CYL/ARC #>>                   |1
   |------------------------------------------------|     DISC BOOTSTRAP
  2|                 READ ORDER                     |2    SIO PROGRAM
   |                                                |     (SYSTEM DISC
  3|               <<MEM ADDRESS>>                  |3    ONLY)
   |------------------------------------------------|
  4|               SIO JUMP ORDER                   |4    Words 0-5 contain
   |                                                |     the ascii string
  5|               <<MEM ADDRESS>>                  |5    "SYSTEM DISC" for
   |------------------------------------------------|     HP-IB Systems
  6|///////////////////|   DISC TYPE    |DISCSUBTYPE|6
   |------------------------------------------------|
  7|                COLD LOAD ID                    |7
   |------------------------------------------------|
 10|         "3"           |          "0"           |8
   |------------------------------------------------|
 11|         "0"           |          "0"           |9    IF WORD %11
   |------------------------------------------------|     CONTAINS A "1"
 12|                                                |10   A FORMER SYSTEM
   |                                                |     VOLUME HAS BEEN
 13|                                                |11   SCRATCHED.
   |                 VOLUME NAME                    |
 14|                                                |12
   |                                                |
 15|                                                |13
   |------------------------------------------------|
 16|                                                |
   .                                                .
   .                                                .
   .                   UNUSED                       .
   .                                                .
   .                                                .
 24|                                                |
   |------------------------------------------------|
 25|                    CYL                         |     ICF WCS
   |------------------------------------------------|     IMAGE
 26|       HEAD               |       SECTOR        |     POINTER
   |------------------------------------------------|
```

System Volume (Cont.)

```
  27|                                              |
    .                                              .
    .               RESERVED                       .
    .                                              .
 122|                                              |
    |----------------------------------------------|
 123|                    CYL                       |
    |----------------------------------------------|
 124|     HEAD                |      SECTOR        |
    |----------------------------------------------|
    .                                              .
    .                                              .
    .                                              .
    .                                              .
    |----------------------------------------------|
 170|                                              |120
    |----------------------------------------------|
 171|          DISC FREE SPACE MAP OK FLAG         |121
    |----------------------------------------------|
 172| DISC FREE SPACE MAP DESCRIPTOR TABLE CHECKSUM |122
    |----------------------------------------------|
 173|  DISC FREE SPACE DESCRIPTOR TABLE DIRTY FLAG |123
    |----------------------------------------------|
 174|                                              |124
    |-- DISC FREE SPACE DESCRIPTOR TABLE ADDRESS --|
 175|                                              |125
    |----------------------------------------------|
 176|                                              |126
    |------- DISC FREE SPACE BITMAP ADDRESS -------|
 177|                                              |127
    |----------------------------------------------|
```

Serial Volume

```
  +-----------------------------------------------+
 0|                                               |0
  |           0  (:STORE)                         |
 1|                                               |1
  |                 or                            |
 2|                                               |2
  |      COLDLOAD SIO CHANNEL PROGRAM (NON-HP-IB  |
 3|      MACHINES ONLY). FOR HP-IB MACHINES, COLD |3
  |      LOAD CHANNEL PROGRAM IS IN SECTOR 2 AND  |
 4|      SOFTDUMP CHANNEL PROGRAM IS IN SECTOR 3. |4
  |                                1  1  1  1  1 1|
 5| 0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5|5
  +--+--+--+--------+----------------+------------+
 6|SC|MV|SR|        |    TYPE        | SUB-TYPE   |6   SC = 1 ==>
  +--+--+--+--------+----------------+------------+      SCRATCH VOLUME
 7|                                               |7   MV = 1 ==> MASTER
  |                                               |      VOLUME OF PV SET.
10|                    0                          |8   SR = 1 ==>
  |                                               |      SERIAL DISC
11|                                               |9
  +----------------------+------------------------+
12|        "S"           |          "E"           |10 \
  +----------------------+------------------------+   |
13|        "R"           |          "D"           |11 | VOL NAME
  +----------------------+------------------------+   |
14|        "I"           |          "S"           |12 | "SERDISC"
  +----------------------+------------------------+   |
15|        "C"           | SDISC VERSION NUMBER   |13 /
  +----------------------+------------------------+
16| WORDS PER SECTOR                              |14 \
  +-----------------------------------------------+   |
17| SECTORS PER TRACK (CARTRIDGE TAPE = 1)        |15 |
  +-----------------------------------------------+   |
20| SECTOR ADDRESS OF BEGINNING OF TAPE (BOT)     |16 |
  +-----------------------------------------------+   | SERIAL
21| DOUBLE ADDRESS OF                             |17 > DISC
  |-                                             -|   | INFO
22| END OF TAPE (EOT)                             |18 |
  +-----------------------------------------------+   |
23| DOUBLE ADDRESS OF                             |19 |
  |-                                             -|   |
24| END OF DATA (EOD)                             |20 /
  +-----------------------------------------------+
25|                   CYL                         |21  ICF WCS
  +----------------------+------------------------+    IMAGE
26|      HEAD            |       SECTOR           |22  POINTER
  +----------------------+------------------------+
```

Serial Volume (Cont.)

```
   +---------------------------------------------+
 27|                                             |23
   .                                             .
   .       RESERVED FOR FUTURE WCS               .
122|                                             |82
   +---------------------------------------------+
123|                  CYL                        |83
   +---------------------+-----------------------+
124|     HEAD            |       SECTOR          |84
   +---------------------+-----------------------+
```

Master Volume

```
                   -------------------------------------------
                  0|                                         |0
                  1|                                         |1
                  2|                  0                      |2
                  3|                                         |3
                  4|                                         |4
                  5|                                         |5
                   -------------------------------------------
SC = SCRATCH      6|SC|MV|SR|    |6   TYPE   11|12  SUB-TYPE 15|6
     VOLUME        -------------------------------------------
MV = MASTER       7|          GENERATION INDEX               |7
  VOLUME = 1       -------------------------------------------
SR = SERIAL      10|                  0                      |8
     VOLUME      11|                                         |9
                   -------------------------------------------
                 12|                                         |10
                 13|               VOLUME                    |11
                 14|                NAME                     |12
                 15|                                         |13
                   -------------------------------------------
                 16|             INITIAL DATE                |14
                   -------------------------------------------
                 17|               DIRBASE                   |15  0 IF NOT
                   -------------------------------------------    MASTER
                 20|               DIRSIZE                   |16  VOLUME
                   -------------------------------------------
                 21|                                         |17
                 22|               ACCOUNT                   |18
                 23|                NAME                     |19
                 24|                                         |20
                   -------------------------------------------
                 25|                                         |21
                 26|                GROUP                    |22
                 27|                NAME                     |23
                 30|                                         |24
                   -------------------------------------------
```

Master Volume (Cont.)

```
                   ---------------------------------------------
                31|                                            |25
                32|               VOLUME SET                   |26
                33|                  NAME                      |27  HEADER
                34|                                            |28
                   ---------------------------------------------
 VS VTAB        35|                                            |29
 HEADER +          ---------------------------------------------
 8 ENTRIES      36|0 VCOUNT 3|           |         VMASK       |30
 COPIED FROM       ---------------------------------------------
 VSET DEFN      37|                                            |31
 IN SYSTEM      40|                VOLUME                      |32
 DIRECTORY      41|                 NAME                       |33  VOLUME
                42|                                            |34  ENTRY 0
                   ---------------------------------------------     .
                43|                                            |35   .
                   ---------------------------------------------     .
                44|  SUB-TYPE            |         VTABX       |36   .
                   ---------------------------------------------     .
                45|                                            |37   .
                  |                      .                     |     .
                                         .                           .
                                         .                          VOLUME
                  |                      .                     |    ENTRY
               116|                                            |78    7
                   ---------------------------------------------
                  .                                            .
                  .                                            .
                  .                                            .
                  .                                            .
                  |--------------------------------------------|
               170|                                            |120
                  |--------------------------------------------|
               171|          Disc Free Space map OK flag       |121
                  |--------------------------------------------|
               172|  DISC FREE SPACE DESCRIPTOR TABLE CHECKSUM |122
                  |--------------------------------------------|
               173| DISC FREE SPACE DESCRIPTOR TABLE DIRTY FLAG|123
                  |--------------------------------------------|
               174|                                            |124
                  |-- DISC FREE SPACE DESCRIPTOR TABLE ADDRESS -|
               175|                                            |125
                  |--------------------------------------------|
               176|                                            |126
                  |------- DISC FREE SPACE BITMAP ADDRESS ------|
               177|                                            |127
                  |--------------------------------------------|
```

Slave Volume

```
                      ---------------------------------------------
                    0|                                            |0
                    1|                                            |1
                    2|                      0                     |2
                    3|                                            |3
                    4|                                            |
 SC = SCRATCH       5|                                            |
      VOLUME          ---------------------------------------------
 MV = MASTER        6|SC|MV|SR|       |6 TYPE 11,12 SUB-TYPE  15|6
   VOLUME = 0         ---------------------------------------------
 SR = SERIAL        7|              GENERATION INDEX              |7
      VOLUME          ---------------------------------------------
                   10|                      0                     |8
                   11|                                            |9
                      ---------------------------------------------
                   12|                                            |10
                   13|                    VOLUME                  |11
                   14|                     NAME                   |12
                   15|                                            |13
                      ---------------------------------------------
                   16|                INITIAL DATE                |14
                      ---------------------------------------------
                   17|                      0                     |15
                   20|                                            |16
                      ---------------------------------------------
                   21|                                            |17
                   22|                  ACCOUNT                   |18
                   23|                   NAME                     |19
                   24|                                            |20
                      ---------------------------------------------
                   25|                                            |21
                   26|                   GROUP                    |22
                   27|                    NAME                    |23
                   30|                                            |24
                      ---------------------------------------------
                   31|                                            |25
                   32|                VOLUME SET                  |26
                   33|                   NAME                     |27
                   34|                                            |28
                      ---------------------------------------------
                     .                                            .
                     .                                            .
```

Slave Volume (Cont.)

```
                  .                                            .
                  .                                            .
                  |--------------------------------------------|
               170|                                            |120
                  |--------------------------------------------|
               171|           DISC FREE SPACE MAP OK FLAG      |121
                  |--------------------------------------------|
               172|  DISC FREE SPACE DESCRIPTOR TABLE CHECKSUM |122
                  |--------------------------------------------|
               173| DISC FREE SPACE DESCRIPTOR TABLE DIRTY FLAG|123
                  |--------------------------------------------|
               174|                                            |124
                  |-- DISC FREE SPACE DESCRIPTOR TABLE ADDRESS -|
               175|                                            |125
                  |--------------------------------------------|
               176|                                            |126
                  |------- DISC FREE SPACE BITMAP ADDRESS ------|
               177|                                            |127
                  |--------------------------------------------|
```

Defective Tracks Table (Sector 1 of Disc)
(Not Used On CS-80 Discs)

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
    0|      # OF DEFECTIVE TRACK ENTRIES (N)         |0
     |-----------------------------------------------|
    1|           DEFECTIVE TRACK NUMBER        | DTC |1     120 DEFECTIVE
     |-----------------------------------------------|      TRACKS MAXIMUM
    2|           DEFECTIVE TRACK NUMBER        | DTC |2
     |-----------------------------------------------|
     |                      .                        |
     |                      .                        |
     -                      .                        -

     -                      .                        -
     |                      .                        |
     |                      .                        |
     |-----------------------------------------------|
  167|           DEFECTIVE TRACK NUMBER        | DTC |119
     |-----------------------------------------------|
  170|           DEFECTIVE TRACK NUMBER        | DTC |120
     |-----------------------------------------------|
  171|                                               |121
     |                                               |
  172|                                               |122
     |                RESERVED FOR                   |
  173|                 FUTURE USE                    |123
     |                                               |
  174|                                               |124
     |                                               |
  175|                                               |125
     |-----------------------------------------------|
  176|         NEXT AVAILABLE ALTERNATE TRACK        |126
     |-----------------------------------------------|
  177|      LOGICAL DISC PACK SIZE (CYLINDERS)       |127
     |-----------------------------------------------|
               OR # OF TRACKS IF FH DISC
```

DTC	(DEFECTIVE TRACK CODE)
0	suspect
1	suspect alternate
2	deleted
3	reassigned

NOTE: The situation where there are two entries for the same track, n, one having a DTC of 0 (suspect) and the other having a DTC 3 (reassigned) results from a situation where the disc driver could not "read" (unreadable) the address of the particular track.

Defective Sector Table (DSCT -- Sector 1 of Disc)

(the DSCT exists on device type 3 (CS-80) discs)

```
        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
       +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
   0 |      NUMBER OF ENTRIES IN THE TABLE                | 0
       +------------------------------------------------+
  %1 |        INDEX TO THE FIRST ENTRY (6)                | 1
       +------------------------------------------------+
  %2 |                ENTRY SIZE (2)                      | 2
       +------------------------------------------------+
  %3 |      MAXIMUM NUMBER OF ENTRIES (61)                | 3
       +------------------------------------------------+
  %4 |                0 (RESERVED)                        | 4
       +------------------------------------------------+
  %5 |                0 (RESERVED)                        | 5
       +------------------------------------------------+
  %6 |        FIRST DEFECTIVE SECTOR ENTRY                | 6
     |     (DOUBLE-WORD LOGICAL SECTOR ADDRESS)           |
       +------------------------------------------------+
 %10 |               SECOND ENTRY                         | 8
     |                                                    |
       +------------------------------------------------+
 %12 |               THIRD ENTRY                          | 10
     |                                                    |
       +------------------------------------------------+
     |                                                    |
     -                                                    -
                         .
                         .
                         .
     -                                                    -
     |                                                    |
       +------------------------------------------------+
%176 |        MAXIMUM DEFECTIVE SECTOR ENTRY              | 126
%177 |                                                    | 127
       +------------------------------------------------+
```

Unlike the DTT, entries in the DSCT are not permanent. Once a suspect sector is handled by INITIAL or VINIT, its entry is removed from the table. Thus, this table contains only unprocessed suspect sectors.

Reserved Area Bit Map (Sector 4 of the System Disc)

The first 400 sectors of the system disc are reserved for Initial's use. This area contains permanent data structures for the boot. It is also used as a temporary storage area for data during sparing. All other system volumes and private volumes reserve only the first 10 sectors of the disc. They do not have a reserved area bit map.

The bit map contains 1 bit per sector. A '1' means the sector is free.

```
      +-----------------------------------------+
  %0  |                                         | 0
      |                                         |
      |             RESERVED AREA               |
      |                BIT MAP                  |
      |                                         |
      |                                         |
      -                                         -
                        .
                        .
                        .
      -                                         -
      |                                         |
 %30  |                                         | 24
      |-----------------------------------------|
 %31  |                                         | 25
      |            RESERVED FOR                 |
      |                FUTURE USE               |
      |                                         |
      -                                         -
                        .
                        .
                        .
      -                                         -
      |                                         |
%177  |                                         | 127
      +-----------------------------------------+
```

Disc Cold Load Information Table (Sectors 28-30)

```
  |-----------------------------------------|
 0|      POINTER TO TABLE INFORMATION       | FREFTR      >--------|
  |-----------------------------------------|                      |
 1|      POINTER TO TEMPORARY CST INFO      | TCSTPTR              |
  |-----------------------------------------|                      |
 2|  # OF ENTRIES TO READ ON DISC COLD LOAD | NREAD                |
  |-----------------------------------------|                      |
 3|  # OF CODE SEGMENTS IN INITIAL          | NVTCST'              |
  |-----------------------------------------|                      |
 4|        INITIAL'S DB VALUE               | INITDB               |
  |-----------------------------------------|                      |
 5|        INITIAL'S DL VALUE               | INITDL               |
  |-----------------------------------------|                      |
 6|        INITIAL'S Z VALUE                | INITZ                |
  |-----------------------------------------|                      |
 7|        INITIAL'S Q VALUE                | INITQ                |
  |-----------------------------------------|                      |
 8|        INITIAL'S S VALUE                | INITS                |
  |-----------------------------------------|                      |
 9|      | SYSDISC TYPE    |    SUBTYPE      | DISCTST              |
  |-----------------------------------------|                      |
10|           COLD LOAD ID                  | COLD'LOAD'ID'        |
  |-----------------------------------------|                      |
11|          LOG FILE NUMBER                | LOG'FILE'NUM'        |
  |-----------------------------------------|                      |
12|          DIRECTORY DISC                 |                      |
  |                                         | DIRADR               |
13|             ADDRESS                     |                      |
  |-----------------------------------------|                      |
14|      LDEV 1   VIRTUAL MEMORY            |                      |
  |                                         | VIRMEMADDR           |
15|          DISC ADDRESS                   |                      |
  |-----------------------------------------|                      |
16|         # LOG PROCS                     |                      |
  |-----------------------------------------|                      |
17|          LOG ID'S                       |                      |
  |-----------------------------------------|                      |
18|          RIN TABLE                      |                      |
  |                                         | RINADR               |
19|          DISC ADDRESS                   |                      |
  |-----------------------------------------|                      |
20|          DIRECTORY SIZE                 | DIRSECT              |
  |-----------------------------------------|                      |
21| #SECTORS IN VIRTUAL MEMORY REGION OF LDEV 1 | SECTORS IN LDEV1VM |
  |-----------------------------------------|                      |
22|              UNUSED                     |                      |
  |-----------------------------------------|                      |
23|          RIN TABLE SIZE                 | RINSECT              |
  |-----------------------------------------|                      |
24|           # OF RINS                     | RINS                 |
  |-----------------------------------------|
```

Disc Cold Load Information Table (Cont.)

```
  |-----------------------------------------|                      |
25|             # of global RINS            | GRINS                |
  |-----------------------------------------|  TL=Tape cold load   |
26|                               |TL|RL|RY| LOAD MODE            |
  |-----------------------------------------|   RL=Reload          |
  |                                         |   RY=recovery        |
27| HIGHEST VOL #        |    # OF VOLUMES  | H'VOL'               |
  |-----------------------------------------|                      |
28|        DISC COLD LOAD ENTRY POINT       | DISCENTRY            |
  |-----------------------------------------|                      |
29|        SYSTEM  DISC DRT NUMBER          | SYSDISCDRT           |
  |-----------------------------------------|                      |
30|        JOB MASTER TABLE                 |                      |
  |                                         | JMATLOC              |
31|        DISC ADDRESS                     |                      |
  |-----------------------------------------|                      |
32|                                         |                      |
  |        IDD DISC ADDRESS                 | IDDLOC               |
33|                                         |                      |
  |-----------------------------------------|                      |
34|                                         |                      |
  |        ODD DISC ADDRESS                 | ODDLOC               |
35|                                         |                      |
  |-----------------------------------------|                      |
36|        WELCOME MESSAGE (DST 47          |                      |
  |                              10)        | LOGONLOC1            |
37|        DISC ADDRESS                     |                      |
  |-----------------------------------------|                      |
38|        WELCOME MESSAGE (DST 48          |                      |
  |                              10)        | LOGONLOC2            |
39|        DISC ADDRESS                     |                      |
  |-----------------------------------------|                      |
40|                                         |                      |
  |     LOG ID ADDRESS                      |                      |
41|                                         |                      |
  |-----------------------------------------|                      |
42|                                         |                      |
  |     LOG TAB ADDRESS                     |                      |
43|                                         |                      |
  |-----------------------------------------|                      |
44|     LOG ID SIZE                         |                      |
  |-----------------------------------------|                      |
45|     LOG TAB SIZE                        |                      |
  |-----------------------------------------|                      |
```

Disc Cold Load Information Table (Cont.)

```
|---------------------------------------------|                   |
|             SIZE IN WORDS                    | FAEFTR+0  <--------|
|----------------------------------           |
|                                   *DRIVER    |
|             MEMORY ADDRESS                   |
|                                    TABLE     |
|----------------------------------           |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             SIZE IN WORDS                    | FAEFTR+5
|----------------------------------           |
|                                             |
|             MEMORY ADDRESS         *CTAB0    |
|                                             |
|----------------------------------           |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             SIZE IN WORDS                    | FAEFTR+10
|----------------------------------           |
|                                             |
|             MEMORY ADDRESS         *CTAB     |
|                                             |
|----------------------------------           |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             SIZE IN WORDS          *         | FAEFTR+15
|---------------------------------- COMMUNICA-|
|                                   TION SUB- |
|             MEMORY ADDRESS         SYSTEM    |
|                                   DRIVER    |
|---------------------------------- TABLE     |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             SIZE IN WORDS          *         | FAEFTR+20
|---------------------------------- COMMUNICA-|
|                                   TION SUB- |
|             MEMORY ADDRESS         SYSTEM    |
|                                   DEFINITION|
|---------------------------------- TABLE     |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
```

Disc Cold Load Information Table (Cont.)

```
|---------------------------------------------|
|             SIZE IN WORDS                    | FAEFTR+25
|----------------------------------           |
|                                   COMMUNICA-|
|             MEMORY ADDRESS         SUBSYSTEM |
|                                   TABLE     |
|----------------------------------           |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             SIZE IN WORDS                    | FAEFTR+30
|----------------------------------           |
|                                   LOGICAL-  |
|             MEMORY ADDRESS         PHYSICAL  |
|                                   DEVICE    |
|---------------------------------- TABLE     |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             SIZE IN WORDS                    | FAEFTR+35
|----------------------------------           |
|                                   LOGICAL-  |
|             MEMORY ADDRESS         DEVICE    |
|                                   TABLE     |
|----------------------------------           |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             SIZE IN WORDS                    | FAEFTR+40
|----------------------------------           |
|                                   DEVICE    |
|             MEMORY ADDRESS         CLASS     |
|                                   TABLE     |
|----------------------------------           |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             SIZE IN WORDS                    | FAEFTR+45
|----------------------------------           |
|                                   VOLUME    |
|             MEMORY ADDRESS         TABLE     |
|                                             |
|----------------------------------           |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
```

Disc Cold Load Information Table (Cont.)

```
|---------------------------------------------|
|             SIZE IN WORDS                    | FAEFTR+50
|----------------------------------           |
|                                   LOGICAL   |
|             MEMORY ADDRESS         DEVICE    |
|                                   TABLE     |
|---------------------------------- EXTENSION |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             STACK SIZE                       | FAEFTR+55
|----------------------------------           |
|                                   INITIAL's |
|             MEMORY ADDRESS         STACK     |
|                                             |
|----------------------------------           |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             SIZE IN WORDS                    | FAEFTR+60
|----------------------------------           |
|                                   DEVICE    |
|            MEMORY ADDRESS          CLASS     |
|                                   TABLE     |
|---------------------------------- HEADER    |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             SIZE IN WORDS                    | FAEFTR+65
|----------------------------------           |
|                                   TERMINAL  |
|            MEMORY ADDRESS          DESCRIPTOR|
|                                   TABLE     |
|----------------------------------           |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|             SEGMENT SIZE                     | FAEFTR+70
|----------------------------------           |
|                                   INITIAL/  |
|             MEMORY ADDRESS         SYSDUMP   |
|                                 COMMUNICATION|
|---------------------------------- RECORD    |
|                                             |
|             DISC ADDRESS                     |
|                                             |
```

Disc Cold Load Information Table (Cont.)

```
|---------------------------------------------|
|             SEGMENT SIZE                     | FAEFTR+75
|----------------------------------           |
|                                   INITIAL's |
|             MEMORY ADDRESS         SEGMENTS  |
|                                             |
|----------------------------------           |
|                                             |
|             DISC ADDRESS                     |
|                                             |
|---------------------------------------------|
|                                             |
.                                             .
.                                             .
.         (MORE SEGMENTS OF INITIAL)           .
.             ININ                            .
|---------------------------------------------|
```

INITIAL Program CST Map

```
LOGICAL         PHYSICAL
 CST#             CST#          SEGMENT NAME

  0                1             ININ       \
  1                2             BOOTSTRAP |----> core resident
  2                3             RESIDENT  /
  3                4             MAINSEG1  \
  4                5             MAINSEG1A |
  5                6             CONFIGURE |       |noncore resident
  6                7             DEFCTRACKS|       |but present in core
  7               10             SETUP     |-------|at completion of
 10               11             TAPEIO    |       |cold load
 11               12             FILEIO    |
 12               13             DISCSPACE /
 13               14             DIRECTORY1
 14               15             DIRECTORY2
 15               16             SL PROGRAM
 16               17             PROCESS
 17               20             MAINSEG1B
 20               21             MAINSEG2
 21               22             MAINSEG3
 22               23             MAINSEG4
```

*code segment swapping starts at completion of MAINSEG1

SYSDUMP/Initial Communication Record

0	MIT VERSION
1	MIT UPDATE
2	MIT FIX
3	VERSION
4	UPDATE
5	FIX
6	EXP SYSTEM NR.
7	HIGHEST DRT
8	HIGHEST LDEV
9	HIGHEST VOL/# OF VOLS
10	# OF ADD'L DRIVERS
11	COLD LOAD COUNT
12	FILES DUMPED
13	SERIAL DISC LOAD
14	TAPE RECORD SIZE
15	DISC COLD LOAD ENTRY
16	MAX INITIAL SEG SIZE
17	SPARE
18	SPARE
19	SPARE
20	DEV CLASS TAB SIZE
21	TERM DESCRIPTOR SIZE
22	OLD VTAB SIZE
23	OLD INFO SIZE
24	CS TABLE SIZE

SYSDUMP/Initial Communication Record (Cont.)

25	SPARE
26	SPARE
27	SPARE
28	SPARE
29	SPARE
30	CONVERSION BITS WORD 1
31	CONVERSION BITS WORD 2
32	CONVERSION BITS WORD 3
33	CONVERSION BITS WORD 4
34	SPARE
35	SPARE
36	SPARE
37	SPARE
38	SPARE
39	SPARE
40	LOG FILE NUMBER

Cold Load Information Table Extension

The Cold Load Information Table Extension is a part of the Cold Load Information Table that has no use in booting the system. It exists for different system level processes to hold information that would only be created during a RELOAD. A good example of this is the system log file number. This is only created on a RELOAD, and changed whenever a log file is full or a boot (other than a RELOAD) is performed.

In order to protect the Cold Load Info Table, the extension was created. In this way NO I/Os should be performed to the Cold Load Information Table during MPE operation. However to process data into the Cold Load Info Extension a process must use the access routine "PROCESS'COLD'LOAD'INFO". The exact calling sequence can be found in KERNELD.

The Cold Load Information Extension is 2 sectors long and immediately follows the SYSDUMP/Initial Communication Record starting at sector address #31 on logical device 1.

The assigned entries are as follows:

```
--------------------------------------------------------
|                                                      |  0
|-----------                            ---------------|
|                                                      |
|-----------                            ---------------|
|             RESERVED FOR FUTURE SYSTEM USE           |  2

|----------                             ---------------|
|                                                      | 20
|------------------------------------------------------|
|  SYSTEM LOGGING FILE NUMBER                          | 21
|------------------------------------------------------|
|  NETWORK MANAGEMENT LOGGING FILE NUMBER              | 22
|------------------------------------------------------|
|  NETWORK MANAGEMENT TRACE FILE NUMBER                | 23
|------------------------------------------------------|
|  FULL/PARTIAL COMMAND DUMP DATE                      | 24
|------------                           ---------------|
|                                                      | 25
|------------------------------------------------------|
|                                                      | 26
|------------                           ---------------|
|            NOT CURRENTLY ASSIGNED                    | 27
|------------                           ---------------|
|                                                      | 28
|------------                           ---------------|

|------------                           ---------------|
|                                                      | 255
--------------------------------------------------------
```

Virtual Disc Space Management Structures

Disc space for data segments is allocated from reserved regions of system volumes which have been assigned the virtual memory supporting (VMS) attribute. The data structure used for accounting and management of the virtual disc space of the various VMS volumes is the Virtual Disc Space Table (VDSMTAB). This structure consists of a circular list of entries, one for each VMS volume. Each entry contains the information defining the state of the virtual memory region on that volume.

Virtual Disc Space Management Table

```
VDSMTAB DST# = 39 (%47)
VDSMTABPTR   = Absolute(%1026) = SYSGLOB %26
```

General Structure

```
                 +------------------+
  %1026------>|  # VMS VOLUMES   |
                 |  FIRST TO LOOK AT|--+
                 +------------------+  |
       +------->|                  |  |
       |    +---|  NEXT IN LIST    |  |
       |    |   |                  |  |
       |    |   +------------------+  |
       |    +-->|                  |<-+
       |    +---|  NEXT IN LIST    |
       |    |   |                  |
       |    |   +------------------+
       |    +-->|                  |
       +--------|  NEXT IN LIST    |
                 |                  |
                 +------------------+
```

VDSMTAB Entry 0 Format

```
           0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
VDSMTAB00|               #WORDS IN VDSMT                 |TABLELENGTH
          |-----------------------------------------------|
VDSMTAB01| # SYSTEM VOLUMES WHICH HAVE VIRTUAL MEMORY    |VMSVOLUMECNT
          |-----------------------------------------------|
VDSMTAB02|     INDEX OF NEXT ENTRY TO ALLOCATE FROM      |STARTENTRY
          |-----------------------------------------------|
VDSMTAB03|            VM PAGE SIZE (512)                 |VMPAGESIZE
          |-----------------------------------------------|
VDSMTAB04|           # SECTORS/VM PAGE (4)               |SECTORSPERVMPAGE
          |-----------------------------------------------|
VDSMTAB05|      OFFSET FROM ENTRY TO BITMAP (%20)        |OFFSETTOBM
          |-----------------------------------------------|
VDSMTAB06|    TOTAL # VM PAGES CONFIGURED IN SYSTEM      |
          |-----------------------------------------------|
VDSMTAB07| LEAST # OF VM PAGES THAT HAVE EVER BEEN AVAIL.|
          |-----------------------------------------------|
          |                                               |


          | VDSMTAB  %10-%17  UNASSIGNED                  |
          |                                               |
          |-----------------------------------------------|
```

VDSMTAB General Entry Format

```
           0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
Word 0    |    INDEX OF NEXT ENTRY IN CIRCULAR LIST        | NEXTINLIST
          |------------------------------------------------|
Word 1    |                    LDEV#                       | LDEV
          |------------------------------------------------|
Word 2    |          STARTING SECTOR OF DEVICE'S           | HOSTARTSECTOR
          |------------------------------------------------|
Word 3    |             VIRTUAL MEMORY REGION              | LOSTARTSECTOR
          |------------------------------------------------|
Word 4    |             # SECTORS IN DEVICE'S              | TOTAL SECTOR
          |------------------------------------------------|
Word 5    |             VIRTUAL MEMORY REGION              | COUNT
          |------------------------------------------------|
Word 6    |  # PAGES IN DEVICE'S VIRTUAL MEMORY REGION     | TOTAL PAGECNT
          |------------------------------------------------|
Word 7    | # OF PAGES AVAILABLE IN DEVICE'S VM REGION     | PAGESAVAILABLE
          |------------------------------------------------|
Word %10|    # OF VALID WORDS IN DEVICE'S BIT MAP        | BMLENGTH
          |------------------------------------------------|
Word %11|        SIZE OF SMALLEST RECENT MISS            |SMALLESTMISS
          |------------------------------------------------|
WORD %12|    SMALLEST NUMBER OF PAGES EVER AVAILABLE     |
          |------------------------------------------------|
%13-%20   |           UNASSIGNED                           |
          |------------------------------------------------|
          |    DEVICE'S VIRTUAL MEMORY BIT MAP             |
          |------------------------------------------------|
          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |
          |------------------------------------------------|
          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |
          |------------------------------------------------|

          ***COMMENT:  A bit on in a device's VMBIT MAP
                       ==> Corresponding VM page is free.
```

Volume Table

```
                                                  SIR #22=%26
                                                  DST #29=%35

                     zero entry
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
word|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
    |     # OF ENTRIES       |                       |
   0|  (NOT COUNTING ZERO)   |  ENTRY SIZE=16(8)     |0
    |------------------------------------------------|
   1|                COLD LOAD ID                    |1
    |------------------------------------------------|.
   2|                 SYSVOLNUM                      |
    |------------------------------------------------|.
   3|       VIRTUAL MEMORY INTEGRITY NUMBER          |
    |------------------------------------------------|
    .                                                .
    .                                                .
    |------------------------------------------------|
  15|////////////////////////////////////////////////|13
    |------------------------------------------------|
```

Typical Private Volume Entry

```
    |--------------------------------------------|
   0|                                            |0    INDEXED BY
    |                                            |     VOLUME #
   1|                                            |1
    |                 VOLUME                     |
   2|                  NAME                      |2
    |                                            |
   3|                                            |3
    |--------------------------------------------|
   4|                                            |4
    |                                            |
   5|                                            |5
    |                  GROUP                     |
   6|                  NAME                      |6
    |                                            |
   7|                                            |7
    |--------------------------------------------|
  10|                                            |8
    |                                            |
  11|                                            |9
    |                 ACCOUNT                    |
  12|                  NAME                      |10
    |                                            |
  13|                                            |11
    |--------------------------------------------|
    |  LOGICAL DEVICE #      |         |VMS|UN|NS|SC|  NS - NON-SYSTEM
  14| (=0 IF NOT MOUNTED)    |         |   |  |  |  |       DOMAIN
    |--------------------------------------------|  SC - SCRATCH
    |          |VSET VTABX |      MVTABX         |  UN - UNREADABLE/
  15|          |           |                     |       UNFORMATTED
    |--------------------------------------------|
```

Typical System Volume Entry

```
   |------------------------------------------------|
  0|                                                |0    INDEXED BY
   |                                                |     VOLUME #
  1|                                                |1
   |                    VOLUME                      |
  2|                     NAME                       |2
   |                                                |
  3|                                                |3
   |------------------------------------------------|
  4|                                                |4
   |                                                |
  5|                                                |5
   |                   0                            |
  6|                                                |6
   |                                                |
  7|                                                |7
   |------------------------------------------------|
 10|   STARTING SECTOR OF VOLUME'S VM (0 IF NONE)   |8
   |                                                |
 11|                                                |9
   |------------------------------------------------|
 12|                                                |10
   |   NUMBER OF SECTORS RESERVED FOR VM ON VOLUME  |
 13|               (0 if none)                      |11
   |------------------------------------------------|
   |   LOGICAL DEVICE #    |          |VMS|UN|NS|SC|   NS - NON-SYSTEM
 14|  (=0 IF NOT MOUNTED)  |          |   |  |  |  |        DOMAIN
   |------------------------------------------------|   SC - SCRATCH
   |           |VSET VTABX |      MVTABX            |   UN - UNREADABLE/
 15|           |           |                        |        UNFORMATTED
   |------------------------------------------------|  VMS - VIRTUAL MEMORY
                                                             SUPPORTING
```

CHAPTER 4 DIRECTORY

Introduction to the Directory

SYSGLOB cells:

```
    DIRBASE <----absolute disc addr of base [SYSGLOB+%130 AND %131]
```

Directory on disc consists of a contiguous area:

```
                |------------------|
   DIRBASE ->   |                  |  The bitmap defines the available/used
                |    DIRECTORY     |  sectors in the directory.  If the
                     BITMAP           directory is <= 6112 sectors, then
                |                  |  the bitmap will occupy 3 sectors.
                |                  |  If the directory size is > 6112
                |------------------|  sectors, then the bitmap will occupy
 DIRBASE+3 ->   |                  |  32 sectors with DIRBASE pointing to
                |    DIRECTORY     |  the 30th sector of the bitmap.  A
                |      DATA        |  zero bit in the bitmap represents a
                |                  |  used sector.  Words 0 and 1 of the
                |     Entries      |  bitmap are ignored.
                |       and        |
                |     Indices      |
                |                  |
                |                  |  Directory entries contain pointers
                |        .         |  which are sector displacements
                         .            relative to DIRBASE.  Entries and
                |        .         |  indices are grouped into "blocks".
                |                  |
                |------------------|
```

The capacities for accounts/groups/users/files are dependent on their block sizes.

*	SYSSAIBSIZE	System acct index block size (3 sectors)
	SYSAUIBSIZE	Acct. user index block size (1-3 sectors)
	SYSAGIBSIZE	Acct. group index block size (1-3 sectors)
	SYSGFIBSIZE	Group file index block size (2 sectors)
	SYSGVSIBSIZE	Group volume set definition ind. blk. size(1 sector)
*	SYSAEBSIZE	Acct. entry block size (3 sectors)
	SYSUEBSIZE	User entry block size (2 sectors)
	SYSGEBSIZE	Group entry block size (2 sectors)
	SYSFEBSIZE	File entry block size (2 sectors)
	SYSVSEBSIZE	Volume set definition entry block size (1 sector)
	SYSMAXBSIZE	Maximum of above. (used to initialize DDS.)

*These values are used once for the creation of the (root) system, account index or new systems. This root index is always at address DIRBASE+3.

Overview of Directory

```
                            ------------
                            | SYSTEM   |
                            ------------
                           /    /   \   \
                          /    /     \   \
                         /    /       \   \ . . .
ACCOUNTS              ----- -----
                      |   | |   |
                      |   | |   |
                      ----- -----
                           /   \
                          /     \
                         /       \
                      -------   --------
                      |USERS|   |GROUPS|
                      -------   --------
                      / /  \    /  \  \
                     / /    \  /    \  \
                  ---- ----  ..  ---- ---- ----
                  | || |         | |  | |  | |  GROUPS
                  | || |         | |  | |  | |
                  ---- ----      ---- ---- ----
                 \          /          /  \
                  ----------          /    \
                    USERS            /      \
                               --------- ---------
                               | VSETS | | FILES |
                               --------- ---------
                                 /      /    /    \
                                /      /    /      \
                               /      /    /        \
                            ----    ---- ----        ----
                   VSETS/   |  |    |  | |  |        |  | FILE
KEY:               VCLASSES |  |    |  | |  |        |  | POINTERS
----------  ----            ----    ---- ----        ----
|        |  |  |                          /
----------  |  |                         /
  INDEX     ----                        /
            ENTRY                   FILE
```

Overview of Directory

Directory Data Segment

```
       0  |--------------------------| 0
       .  |          SECTOR          | .
       .             BUFFER            .
       .          128(10) WORDS        .
     177  |                          | 127
          |--------------------------|
     200  | ADJUST (DB-DL)           | 128
          |--------------------------|
     201  | XTYPE (INPUT PARM)       | 129
          |--------------------------|
     202  |             :   XMVTABX  | 130
          |--------------------------|
     203  | XINDEXP (FINAL INDEX PRT)| 131
          |--------------------------|
     204  | XANAME (DB REL ADDR)     | 132
          |--------------------------|
     205  | XGUNAME (DB REL ADDR)    | 133
          |--------------------------|
     206  | XFNAME (DB REL ADDR)     | 134
          |--------------------------|
     207  | XASEC (ACCOUNT SECURITY) | 135
          |--------------------------|
     210  |                          | 136
          |-XGSEC (GROUP SECURITY)  -|
     211  |                          | 137
          |--------------------------|
     212  | SIRRETURN (FROM GETSIR)  | 138
          |--------------------------|
 213-240    DIRECTORY POINTER "A"      139-160 \
          |--------------------------|          > SEE Directory
 241-266    DIRECTORY POINTER "B"      161-182 /  Pointer Area
          |--------------------------|
     267  | SYS.ACCT.INDEX BLOCK SIZE| 183
          |--------------------------|
     270  |   LDEV    :  DIRECTORY   | 184
          |-------------            -|
     271  | PV DIRECTORY SIZE        | 185
          |--------------------------|
          | PRIVATE VOLUME DIR. SIZE | 186
          |--------------------------|
          |//////////////////////////| 187
          |--------------------------|
          |//////////////////////////| 188
          |--------------------------|
          |//////////////////////////| 189
          |--------------------------|
          |//////////////////////////| 190
          |--------------------------|
          |//////////////////////////| 191
          |--------------------------|
```

Directory Data Segment (Cont.)

```
                 |--------------------------|
                 |//////////////////////////|  192
                 |--------------------------|
                 |//////////////////////////|  193
                 |--------------------------|
                 |//////////////////////////|  194
                 |--------------------------|
                 |//////////////////////////|  195
                 |--------------------------|
                 |//////////////////////////|  196
                 |--------------------------|
                 |//////////////////////////|  197
                 |--------------------------|
             306 |       DISTRIBUTION       |  198
GOODPERCENT=.85  |-                        -|
             307 |          FACTOR          |  199
                 |--------------------------|
             310 |          BASE            |  200
                 |--------------------------|  ---
             311 |                          |  201
                           DA AREA             DDSBWSIZE
                 |                          |
                 |--------------------------|  ---


                 |--------------------------|  ---
                 |         WORK AREA        |
                   (SIZE OF LARGEST ENTRY)     MAX
                 |                          |
                 |--------------------------|  ---


                 |--------------------------|  ---
            1145 |                          |  613
                             DB AREA           DDSBWSIZE
                 |                          |
                 |--------------------------|  ---
```

Directory Pointer Area [DA or DB] DST=20(10) SIR=8(10)

```
^ |--------------------------------|
| |   LDEV        | DIRECTORY BASE| 139/161 DIRBASE1'
| |---------------                 |
| |  ADDRESS OF PAGE IN BUFFER     | 140/162 DIRBASE2'
| |--------------------------------|
| | DIRECTORY PAGE IN BUFFER       | 141/163 CONTENTS
| |--------------------------------|
| | DB ADDRESS OF 1ST ELEMENT      | 142/164 LPNTR
| |--------------------------------|
| | STARTING ADDRESS OF BUFFER     | 143/165 IOPNTR
| |--------------------------------|
| | # VALID PAGES IN BUFFER        | 144/166 NUMVALID
| |--------------------------------|
| | D|                          |B | 145/167 D=DIRTY FLAG, B=BAD ELEMENT
| |--------------------------------|
| | ELEMENT SIZE                   | 146/168 XSIZE      NOTE:
| |--------------------------------|
**| # WORDS USED IN BLOCK          | 147/169 USED    **  INDEXES AND
| |--------------------------------|                     ENTRIES
| | BLOCK SIZE (SECTORS)           | 148/170 BSIZE
| |--------------------------------|                  *  INDEXES ONLY
| | BLOCK SIZE (WORDS)             | 149/171 BWSIZE
| |--------------------------------|
| | MAX # ELEMENTS/BLOCK           | 150/172 BFACTOR
| |-|-|-|-|----------|-------------|
| |I|P| TY|ELEMENT SIZE|BLOCK SIZE| 151/173 MISCWD
| | | | | |   (WORDS)  | (SECTORS)|
| |-|-|-|-|------------|-----------|
| | NUMBER OF ELEMENTS             | 152/174 XCOUNT
| |--------------------------------|
| | NUMBER OF ACCESSORS            | 153/175 PCOUNT
| |--------------------------------|
| | ENTRY TOTAL                    | 154/176 ETOTAL
| |-|-|-|-|----------|-------------|
| |O|P| TY|ENTRY SIZE| BLOCK SIZE  | 155/177 EMISCWD
| |-|-|-|-| (WORDS)  | (SECTORS)   |
| |-|-|-|-|----------|-------------|
| | FATHER INDEX POINTER           | 156/178 PINDEXP
| |--------------------------------|
| |  F             |               | 157/179
* |----A-----------|---------------|
| |      T         |   N           | 158/180 PNAME   TY = 0-FILE
| |--------H-------|-----A---------|                      1-GROUP
| |          E     |     M         | 159/181              2-ACCT
| |------------R---|---------E-----|                      3-USER
| |                |               | 160/182              4-VSD
v |----------------|---------------|                 I  = 0-ENTRY BLOCK
                                                          1-INDEX BLOCK
                                                     P  = PURGE FLAG
```

Directory Space Data Segment (DIRSDS)

```
DST=21 (%25)

SIR=8
    10
```

```
                 DST = 21 ( %25 )

                           1 1 1 1 1 1
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
      |===============|===============|
  0   | Logical device|  Bit map      |
      |---------------|               |
  1   |     base sector address       |  DS'BASE
      |-------------------------------|
  2   | Ptr to last avail word in buff|  DS'LAST'WORD
      |-------------------------------|
  3   | Ptr to first word in buffer   |  DS'FIRST'WORD
      |-------------------------------|
  4   | Size in sectors of directory  |  DS'DIR'SIZE
      |-------------------------------|
  5   |D|E|S|P|                       |  DS'FLAGS
      |-------------------------------|
  6   | First current sector in buff  |  DS'CUR'SECTOR
      |-------------------------------|
  7   |  Disc address of current part |
      |                               |  DS'ADDR
 10   |    of bit map in the buffer   |
      |-------------------------------|
 11   |    Size of buffer in words    |  DS'SIZE
      |-------------------------------|
 12   |     Next requested sector     |  DS'REQ'SECTOR
      |-------------------------------|
 13   |    Last sector in bit map     |  DS'LAST'SECTOR
      |-------------------------------|
 14   |  System saved pntr to last    |  DS'SYS'LAST
      |-------------------------------|
 15   |  System saved pntr to first   |  DS'SYS'FIRST
      |-------------------------------|
 16   |  System saved current sector  |  DS'SYS'CUR
      |-------------------------------|
 17   |     Saved directory size      |  DS'SYS'SIZE
      |-------------------------------|
 20   | LDEV that last error occurred |  DS'ERROR'LDEV
      |-------------------------------|
 21   | Type of error that occurred   |  DS'ERROR'TYPE
      |-------------------------------|
```

```
|                              |
| This section of the bit map  |
| DST is occupied by up to 3   |
| sectors of bit map.  It is   |
| swapped in 3 sectors at a    |
| time as needed. DS'FIRST'WORD|
| is updated to search for     |
| space in the bit map.  When  |
| it reaches DS'LAST'WORD for  |
| the second pass, the next 3  |
| sectors of bit map will be   |
| swapped in.                  |
|                              |
|==============================|
```

Partial definitions:

```
DS'LDEV         =  DS'BASEO.(0:8)
DS'DIRTY        =  DS'FLAGS.(0:1)
DS'ERR'IN'PROG =  DS'FLAGS.(1:1)
DS'DIR'DISABLED=  DS'FLAGS.(2:1)
DS'PERM'DISABLE=  DS'FLAGS.(3:1)
```

Descriptions:

DS'ADDR

This is the address of the section of bit map that is currently in the buffers. For example, this address will usually be the same as DS'BASE. If we need to page in more sectors of bit map than the first three, then this address will be subsequently larger than DS'BASE.

DS'BASE

This is the base address of the directory bit map. If the directory is greater than 6112 sectors, then this address will be 29 sectors less than the address found in the Cold Load Information table on disc.

DS'CUR'SECTOR

This is the current bit map sector number of the first sector in the buffer area. Its value can range from 1 to 30. This number minus one added to DS'BASE will result in DS'ADDR.

DS'DIR'DISABLED

If this bit is on, the directory allocation and deallocation is off and only a WARMSTART will turn this bit off. The bit is turned on if an I/O error occurs on a directory bit map sector or if we find data integrity problems with the bit map, i.e. if we attempt to deallocate a sector that is already deallocated.

DS'DIR'SIZE

This is the size (sectors) of the directory area. This size includes only the last 3 sectors of the bit map. If the directory is greater than 6112 sectors, then this size does not include the extra 29 sectors of bit map. It can also be thought of as the number of bits in the bit map.

DS'DIRTY

This bit is set if the bit map sectors in the buffer have been modified in any way. When more sectors must be brought into the buffers, or if we switch to a different domain (system to PV, PV to system) this bit is interrogated to determine if the sectors presently in the buffers must be first written to disc.

DS'ERROR'LDEV

The LDEV in which the last directory error occurred.

DS'ERROR'TYPE

This word describes the type of directory bit map error that occurred. Its legal values are:

0 - No error
1 - I/O error on a write
2 - I/O error on a read
3 - Attempting to deallocate space that is already deallocated
4 - Directory space management is already disabled

DS'ERR'IN'PROGRESS

A directory space management error is currently in progress.

DS'FIRST'WORD

A DST relative pointer to the word in the bit map buffer that we will interrogate next when directory space is needed. When the system first comes up, this word is always initialized to DS'HEADER+2 (i.e. to point to the first word in the bit map). On subsequent bit map sector reads, it is set to DS'HEADER since subsequent sectors will not have the 2 word overhead that exists in the first sector of the bit map.

DS'FLAGS

This word contains numerous flags. See individual descriptions.

DS'LAST'SECTOR

This is the total number of active bit map sectors. This number will range from 1 to 32.

DS'LAST'WORD

This is the current number of bit map word in the buffer. It can range from 1 to %577 + DS'HEADER. If there exists 3 full sectors in the buffer, then it will have the value %600 + DS'HEADER - 1 or %621. It is compared to DS'FIRST'WORD to determine if we have hit the end of the current buffer area.

DS'PERM'DISABLE

If this bit is set, then directory allocation/deallocating is permanently disabled. This bit should not be set.

DS'REQ'SECTOR

This is the next sector to begin reading in up to 3 bit map sectors. It is updated by 2 or 3 and the read procedure will bring in up to 3 sectors starting from this sector. If this sector is set to be greater than DS'LAST'SECTOR, then it is reset to 1. After the sectors are read in, DS'CUR'SECTOR is set the DS'REQ'SECTOR.

DS'SIZE

This is the size in words of the bit map buffer area. It is always a multiple of a sector (128 words). It will usually have the value of %600. Legal values are %200, %400 and %600.

DS'SYS'LAST, DS'SYS'FIRST, DS'SYS'CUR & DS'SYS'SIZE

The values of DS'LAST'WORD, DS'FIRST'WORD, DS'CUR'SECTOR and DS'SIZE will be stored in these locations when the directory space management switches from the system directory to a private volume directory. And, of course, when DSM switches back to system domain, the above mentioned values are reinitialized with these values.

Directory Structure

```
                                        INDEX BLOCK
                                        -----------

                          --------> |--------------------|
                          |         | INDEX PREFIX       |
                          |         |--------------------|
                          |         | INDEX PREFIX       |
                          |         |--------------------|
                          |         |                    |
                          |         |--------------------|
                          |              ENTRY BLOCKS
                          |   -----> |--------------------|
                          |   |      |       ENTRY        |
                          |   |      |--------------------|
                          |   |      |       ENTRY        |
                          |   |      |--------------------|
                          |   |      |       ENTRY        |
                          |   |      |--------------------|
                          |   |
                          |   |   --> |--------------------|
    INDEX BLOCK           |   |   |   |       ENTRY        |
    -----------           |   |   |   |--------------------|
|--------------------|    |   |   |   |       ENTRY        |
|       INDEX        |    |   |   |   |--------------------|
|       BLOCK        |-----   |   |   |       ENTRY        |
|       PREFIX       |        |   |   |--------------------|
|--------------------|        |   |   |       ENTRY        |
|       INDEX        |        |   |   |--------------------|
|       BLOCK        |---------   |   |       ENTRY        |
|       ENTRY        |            |   |--------------------|
|--------------------|            |   |       ENTRY        |
|       INDEX        |            |   |--------------------|
|       BLOCK        |-------------
|       ENTRY        |
|--------------------|
```

The Index Block prefix points back to
the previous higher level. The Index
Block entries point to the entry blocks.

Directory Definitions

```
>PAGE    - smallest allocatable record ("phys.recd")-currently sector.
>BLOCK   - integral# of pages; contains contiguous indices or entries.
>INDEX   - pointer to entry block, containing name of 1st entry.
>ENTRY   - information-containing "object" may contain pointer to an
           index block.
>POINTER - 15-bit positive relative page number (relative to directory
           base).
>DDS     - directory data segment.
>ELEMENT - a generic name for index or entry.
```

Index Block Prefix (10 Words)

```
  0-FILE \
  1-GROUP|  3 bits       ------------>| INDEX SIZE (WORDS)
  2-ACCT |<--------      |            | 7 BITS
  3-USER /        |      | *
  4-VSET          |      |
                  |      |      ----->| BLOCK SIZE (SECTORS)
PURGE FLAG<----   |      |      |     | 4 BITS
              |   |      |      |
           |-|-|----|----------|-----|
MISCWD    0|1|P| TY |  XSIZE   |BSIZE|0  INDEX BLOCK INFO.
           |-|-|----|----------|-----|
          1|        XCOUNT           |1  NUMBER OF INDEX POINTERS
           |-------------------------|
          2|       IPCOUNT           |2  NUMBER OF ACCESSORS*
           |-------------------------|
          3|        ETOTAL           |3  ENTRY TOTALS
           |-|-|----|---------|------|
EMISCWD   4|0|P| TY | EXSIZE  |EBSIZE|4  ENTRY BLOCK INFO.
           |-|-|----|---------|------|
          5|      PINDEXP            |5  INDEX POINTER OF FATHER
           |-------------------------|
          6|                         |6 \
           |                         |  |
          7|                         |7 |
           |        PNAME            |   >NAME OF FATHER
         10|                         |8 |
           |                         |  |
         11|                         |9 /
           |=========================|
           |                         |
           |                         |
```

*The count is incremented by each access that uses and relies upon a pointer to the index block, i.e., it is guaranteed not to be purged while the count is not = 0.

Index Entry (6 Words)

```
|-------------------------|
|                         |0  1st NAME OF ENTRY BLOCK
|                         |
|                         |1
|       IE1STNAME         |
|                         |2
|                         |
|                         |3
|                         |
|-------------------------|
|        IEPNTR           |4  POINTER TO ENTRY BLOCK
|-------------------------|
|        IECOUNT          |5  NUMBER OF ENTRIES IN e BLOCK
|-------------------------|
```

Account Entry (%36 Words)

```
   |-------------------|
  0|                   |0
   |                   |
  1|                   |1  ACCT.NAME
   |      ANAME        |
  2|                   |2
   |                   |
  3|                   |3
   |-------------------|
  4|     AGIPNTR       |4  ACCT.GROUP INDEX POINTER
   |-------------------|
  5|     AUIPNTR       |5  ACCT.USER INDEX POINTER
   |-------------------|
  6|                   |
   |      ACAP         |   CAPABILITY
  7|                   |7
   |-------------------|
 10|                   |8
   |      ALATTR       |   LOCAL ATTRIBUTES
 11|                   |9
   |-------------------|
 12|                   |10 PASSWORD
   |                   |
 13|                   |11
   |      APASS        |
 14|                   |12
   |                   |
 15|                   |13
   |-------------------|
```

Account Entry (Cont.)

```
     |--------------------|
   16|                    |14
     |     ADFSCOUNT      |   DISC FILE SPACE COUNT (SECTORS)
   17|                    |15
     |--------------------|
   20|                    |16
     |     ADFSLIMIT      |   DISC FILE SPACE LIMIT (SECTORS)
   21|                    |17
     |--------------------|
   22|                    |18
     |     ACPUCOUNT      |   CPU TIME COUNT (SECONDS)
   23|                    |19
     |--------------------|
   24|                    |20
     |     ACPULIMIT      |   CPU TIME LIMIT (SECONDS)
   25|                    |21
     |--------------------|
   26|                    |22
     |   ACONTIMECOUNT    |   CONNECT TIME COUNT (MINUTES)
   27|                    |23
     |--------------------|
   30|                    |24
     |   ACONTIMELIMIT    |   CONNECT TIME LIMIT (MINUTES)
   31|                    |25
     |--------------------|
 ---32| |/|//| | | | | | | |26 FLAGS (SEE BELOW)
 |    |-|-|--|-|-|-|-|-|-|-|
 |  33|S|A|/////|          |27 MAX.JOB PRIORITY
 |    |---------|----------|
 |  34|COMM FILE REC # ACCT|28 command file location of       HARD  CODED
 |    |--------------------|   account udcs                     0     1
 |  35|COMM FILE REC # SYS |29 command file location of         |     |
 |    |--------------------|   system udcs (SYS acct only)      |     |
 |                                                              |     |
 |                                                              |     |
 |                                                              |     |
 |      |--|--|--|--|----|--|----|--|----|--|----|--|----|--|----|--|
 |      | P|//|/////|  R | R|  A | A|  W | W|  L | L|  X | X|  S | S|
 |->ASECW|  |//|/////|ANY |AC|ANY |AC|ANY |AC|ANY |AC|ANY |AC|ANY |AC|
        |--|--|--|--|----|--|----|--|----|--|----|--|----|--|----|--|
                               \---------------- -----------------/
                                               |
   P    PURGE flag                             |
                                          FILE SECURITY
```

S   If 1, system level UDC's exist (only in "SYS" account)
A   If 1, account level UDC's exist for account

Group Entry (%51 Words)

```
     |--------------------|
    0|                    |0  GROUP NAME
     |                    |
    1|                    |1
     |       GNAME        |
    2|                    |2
     |                    |
    3|                    |3
     |--------------------|
    4|      GFIPNTR       |4  GROUP FILE INDEX POINTER
     |--------------------|
    5|                    |5
     |                    |
    6|                    |6  PASSWORD
     |       GPASS        |
    7|                    |7
     |                    |
   10|                    |8
     |--------------------|
   11|                    |9  DISC FILE SPACE COUNT (SECTORS)
     |     GDFSCOUNT      |
   12|                    |10
     |--------------------|
   13|                    |11 DISC FILE SPACE LIMIT (SECTORS)
     |     GDFSLIMIT      |
   14|                    |12
     |--------------------|
   15|                    |13 CPU TIME COUNT (SECONDS)
     |     GCPUCOUNT      |
   16|                    |14
     |--------------------|
   17|                    |15 CPU TIME LIMIT (SECONDS)
     |     GCPULIMIT      |
   20|                    |16
     |--------------------|
   21|                    |17 CONNECT TIME COUNT (MINUTES)
     |   GCONTIMECOUNT    |
   22|                    |18
     |--------------------|
   23|                    |19 CONNECT TIME LIMIT (MINUTES)
     |   GCONTIMELIMIT    |
   24|                    |20
     |--------------------|
   25|*P|                 |21 GROUP SECURITY (SEE BELOW)
     |--|      GSEC       |
   26|                    |    *P = PURGE FLAG
     |--------------------|
```

Group Entry (Cont.)

```
     |--------------------|
   27|    GCAPABILITY     |23 GROUP CAPABILITY
     |--------------------|
   30|     GLINKAGE       |24 GROUP DIR. BASE LINKAGE
     |--------------------|
   31|    GVSDIPNTR       |25 GROUP VOL SET DEFN INDX
     |--------------------|
   32|     GHVSNAME       |26 HOME VOL SET NAME
     |-                  -|
   33|                    |27
     |-                  -|
   34|     GHVSANAME      |28 (Definition's acct name)
     |-                  -|
   35|                    |29
     |--------------------|
   36|                    |30
     |-                  -|
   37|                    |31
     |-    GHVSGNAME     -|    (Definition's group name)
   40|                    |32
     |-                  -|
   41|                    |33
     |--------------------|
   42|                    |34
     |-                  -|
   43|                    |35
     |-    GHVSVSNAME    -|    (Definition's vol set name)
   44|                    |36
     |-                  -|
   45|                    |37
     |--------------------|
   46|    GSAVEFIPNTR     |38 SAVE CELL FOR GFIPNTR
     |--------------------|
   47|   GMOUNTREFCNTR    |39 GROUP BIND COUNTER
     |--------------------|
   50|         0          |40 GSPARE
     |--------------------|

                        GLINKAGE

 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
|---------------------------------------------------------------|
|PV |///////////////////////////////|   MVTABX                  |
|---------------------------------------------------------------|
```

Group Entry (Cont.)

```
GLINKAGE   (0:1) = 0; HVS is in System Domain
           (0:1) = 1; HVS is in Private Volume Domain
           (8:8) = 0; If not PV or Not Bound
           (8:8) <>0; If PV and Bound

            GROUP SECURITY MASK
        |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---
        | P |///| R | R | R | R | R | A | A | A | A | A | W | W | W | W
      25|   |///|ANY|AC |AL |GU |GL |ANY|AC |AL |GU |GL |ANY|AC |AL |GU
        |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---
        | W | L | L | L | L | L | X | X | X | X | X | S | S | S | S | S
      26|GL |ANY|AC |AL |GU |GL |ANY|AC |AL |GU |GL |ANY|AC |AL |GU |GL
        |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---
```

File Entry (File Pointer)(6 Words)

```
|--------------------|
|                    |0  FILE NAME
|                    |
|       FNAME        |1
|---                 |
| B|                 |2
|---                 |
|                    |3
|--------------------|
| FVTABINX |         |4  VOL TABLE INDX / FILE LABEL DISC
|----------|         |                        ADDRESS
|    FLABELADDR      |5
|--------------------|
```

B - Bad file label
  (0:1) = 0 - not defective
        = 1 - defective

User Entry (19 Words)

```
        |--------------------|
       0|                    |0  USER NAME
       1|       UNAME        |1
       2|                    |2
       3|                    |3
        |--------------------|
       4|                    |4  CAPABILITY
        |        UCAP        |
       5|                    |5
        |--------------------|
       6|                    |6  LOCAL ATTRIBUTES
        |       ULATTR       |
       7|                    |7
        |--------------------|
      10|                    |8  PASSWORD
      11|       UPASS        |9
      12|                    |10
      13|                    |11
        |--------------------|
      14|                    |12 HOME GROUP (MAY BE NULL)
      15|      UHGROUP       |13
      16|                    |14
      17|                    |15
        |--------------------| LOG CNT (# OF USERS LOGGED ON)
      20|     ULOGCOUNT      |16 INIT TO 1 FOR MANAGER.SYS SO
        |--------------------|   THIS USER CANNOT BE PURGED
UMAXJOBW 21|*P|U|  0  | JOBPRI |17 MAX.JOB PRI;*P=PURGE FLAG
        |----------|---------|              U=UDC EXIST FLAG
      22|COMM FILE REC #     |18
        |(command file loc of|
        | user udcs)         |
        |--------------------|
```

User Attributes/Capability

```
                               /         SAVE FILES  ------
       FILE-ACCESS ATTRIBUTES <                             |
                               \ NON-SHARABLE DEVICES--     |
                                 COMMUNICATIONS------ |     |
                              NODE MANAGER--------|  | |    |
                              NETWORK ADMINISTRATOR  | |    |
            / ----------------SYSTEM MGR         | |  | |   |
            |  |  ------------ACCOUNT MGR        | |  | |   |
            |  |  |  ---------ACCOUNT LIBRN      | |  | |   |
   USER     |  |  |  |  ------GROUP LIBRN        | |  | |   |
            |  |  |  |  |  ---DIAGNOSTICIAN      | |  | |   |
            |  |  |  |  |  |  --SYSTEM SUPVSR    | |  | |   |
   ATTR     |  |  |  |  |  |  |  CREATE VOLS     | |  | |   |
            |  |  |  |  |  |  |  |  USE VOLS    | |  | |   |
            \  |  |  |  |  |  |  |  |  USER LOGGING |  | |   |
               |  |  |  |  |  |  |  |  |  SYSTEM PROCESS HANDLING
               |  |  |  |  |  |  |  |  |     PROGRAMMATIC SESSIONS

               |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
              |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
              |SM|AM|AL|GL|DI|OP|CV|UV|LG|SP|PS|NA|NM|CS|ND|SF|
              |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|


               0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
              |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
              |//|//|//|//|//|//|//|BA|IA|PM|//|//|MR|//|DS|PH|
              |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|


                                        |  |  |        |     |  |
            /       batch access       -|  |  |        |     |  |
            |interactive access        ----|  |        |     |  |
   ACCESS   |   privileged mode        -------|        |     |  |
   TO      <                                           |     |  |
   GENERAL  |      multiple RINS       ----------------|     |  |
   RESOURCES|extra data segment        ----------------------|  |
            \  process handling        -------------------------|
```

Volume Set Definition Entry

```
               |----------------------------------------|
              0|                                        |0
              1|                                        |1  VOLUME
              2|                GVSNAME                 |2  SET
              3|                                        |3  NAME
               |----------------------------------------|
  TY = 0     4|TY|A|2            7|       MVTABX        |4  GVSLINKAGE
               ------------------------------------------
             5|VOL COUNT|4       7|        VMASK        |5  GVSINFO
               ------------------------------------------
          /  6|                                         |6  MEMBER VOLUME
          |  7|                                         |7  NAME(1ST ENTRY
VOLUME    | 10|             GVSVOLUME                   |8  IS MASTER
ENTRY 0   < 11|                                         |9  VOLUME)
(6 WORDS) |   |-----------------------------------------|
          | 12|0                                   14| M|10 GVSVOLFLAGS
              |-----------------------------------------|
          \ 13| PSEUDO SUBTYPE     |        VTABX       |11 GVSVOLINFO
              |-----------------------------------------|
          / 14|                                         |12
VOLUME    | . |                 .                       | .
ENTRIES   | .                   .                         .
1 - 7     < .                   .                         .
          | . |                 .                       | .
          \ 57|                                         |47
              |-----------------------------------------|
            60|                                         |48
              |                                         |
            61|                                         |49
              |                                         |
            62|             GVSVOLUME                   |50 MEM. VOL.
              |                                         |   NAME
            63|                                         |51
              |-----------------------------------------|
            64|  GVSVOLFLAGS   (MEMBER VOLUME FLAGS)    |52
              |-----------------------------------------|
            65|  GVSVOLINFO    (MEMBER VOLUME INFO)     |53
              |-----------------------------------------|
            66|  GVSDREFCNT    (DEFN. REF. CNTR.)       |54
              |-----------------------------------------|
            67|                 0                       |55  SPARE
              |-----------------------------------------|
```

TY = 0 VOLUME SET DEFINITION
   = 1 VOLUME CLASS
MVTABX: MOUNTED VOLUME TABLE INDEX (IF MOUNTED)
VOL COUNT: NO. OF VOLUMES
VMASK: VOLUME MASK
M = 0 NOT MOUNTED
  = 1 MOUNTED
VTABX: VOLUME TABLE INDEX

GVSLINKAGE

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
|---------------------------------------------------------------|
| T | A |            NOT            |            MVTABX         |
|   |   |            USED           |                           |
|---------------------------------------------------------------|
```

T - TYPE
     0 = Volume Set Definition
     1 = Volume Set Class
A - ALLOCATING FLAG
     0 = not initially allocating (not 1st user of set)
     1 = 1st user of set allocating resources (transitional)
MVTABX - Mounted Volume Table Index
        0 if volume set not logically mounted

GVSINFO

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
|---------------------------------------------------------------|
|    VOLCNT     |      NOT      |            VSMASK             |
|               |      USED     |                               |
|---------------------------------------------------------------|
```

VOLCNT - Number of members in set
VSMASK - Bit mask of volume member usage
        Order is from right to left
         i.e., bit 15 is 1st member, bit 14 is 2nd member ...

GVSVOLFLAGS

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
|---------------------------------------------------------------|
|                          NOT USED                         | M |
|                                                           |   |
|---------------------------------------------------------------|
```

M - Member Mounted Flag
     0 = not mounted
     1 = mounted

GVSVOLINFO

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
|---------------------------------------------------------------|
|             DISC              |             VTABX             |
|        PSEUDO SUBTYPE         |                               |
|---------------------------------------------------------------|
```

DISC PSEUDO-SUBTYPE = (Actual type *16) + actual subtype.
VTABX - Volume Table Index

Volume Set Class Entry

```
                1 1 1 1 1 1
   0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
   |-------------------------------|
  0|                               | 0   VOLUME CLASS NAME
   |-                             -|
  1|                               | 1
   |-          GVCNAME            -|
  2|                               | 2
   |-                             -|
  3|                               | 3
   |-------------------------------|
  4|         GVCLINKAGE            | 4   VOLUME CLASS IDENTIFICATION
   |-------------------------------|
  5|          GVCINFO              | 5   VOLUME CLASS INFORMATION
   |-------------------------------|
  6|          GVCPNAME             | 6   PARENT VOLUME SET DEFINITION
   |-                              |
  7|                               | 7
   |-         GVCPANAME           -|     ACCOUNT OF PARENT DEFINITION
 10|                               | 8
   |-                             -|
 11|                               | 9
   |-------------------------------|
 12|                               |10
   |-                             -|
 13|                               |11
   |-         GVCPGNAME           -|     GROUP OF PARENT DEFINITION
 14|                               |12
   |-                             -|
 15|                               |13
   |-------------------------------|
 16|                               |14
   |-                             -|
 17|                               |15
   |-         GVCPVSNAME          -|     VSNAME OF PARENT DEFINITION
 20|                               |16
   |-                             -|
 21                                |17
   |-------------------------------|
 22|              0                |18
   |-------------------------------|
 23|              0                |19


   |-------------------------------|
 67|              0                |55
   |-------------------------------|
```

GVCLINKAGE

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
 |---------------------------------------------------------------|
 | T |   \/////////////////////////////////////////////////////// |
 |---------------------------------------------------------------|
```

T - TYPE
1 = Volume Set Definition
0 = Volume Set Class

GVCINFO

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
 |---------------------------------------------------------------|
 |    VOLCNT     |      NOT      |             VCMASK            |
 |               |      USED     |                               |
 |---------------------------------------------------------------|
```

VOLCNT - Number of members in set
VCMASK - Bit mask of volume member usage (VOLUME CLASS MASK)
Order is from right to left
i.e. bit 15 is 1st member, bit 14 is 2nd member ...

Volume Mask Format

- USED IN MVTAB, PVUSER, FILE CONTROL BLOCK (FCB), VOLUME SET/CLASS DEFINITION, VOLUME SET VTAB.
- 8-BIT MASK.

```
    ---------------------------------------
    | V7 | V6 | V5 | V4 | V3 | V2 | V1 | V0 |
    ---------------------------------------
      ^    ^    ^    ^    ^    ^    ^    ^
      |    |    |    |    |    |    |    |
      |    |    |    |    |    |    |    -- VOLUME 0 (MASTER)
      |    |    |    |    |    |    |
      |    |    |    |    |    |    ------- VOLUME 1
      |    |    |    |    |    |
      |    |    |    |    |    ------------ VOLUME 2
      |    |    |    |    |
      |    |    |    |    ----------------- VOLUME 3
      |    |    |    |
      |    |    |    ---------------------- VOLUME 4
      |    |    |
      |    |    --------------------------- VOLUME 5
      |    |
      |    -------------------------------- VOLUME 6
      |
      ------------------------------------- VOLUME 7
  0:  NOT MOUNTED OR NON-MEMBER    1:   MOUNTED OR MEMBER
```

CHAPTER 5 LOCK RESOURCES

SIR# Allocation DST %53

Sir's Ordered by Sir Number

SIR #	RANK	SIR NAME
1	10	LOAD PROCESS
2	335	CACHE CONTROL
3	91	IDD
4	92	ODD
5	50	PROCESS TREE STRUCTURE
6	60	SCHEDULING QUEUE
7	70	CST ENTRIES
8	80	SYSTEM DIRECTORY
9	90	LPDT
10	85	LDT
11	110	STORAGE IN OVERLAY AREA
13	130	JPCNT
14	140	JCUT
15	27	JMAT
16	5	FMAVT
17	22	LOADER SEGMENT TABLE
18	180	VDD
19	190	SPOOL
20	200	MESSAGE CATALOGUE
21	210	RIT
22	220	VOLUME TABLE
23	230	WELCOME MESSAGE SIR
24	240	ASSOCIATION TABLE
25	250	CS ALLOCATE
26	260	LOGGING BUFFER
27	83	PV MVTAB
28	280	MEASSIR
29	290	PV USER TABLE
30	300	IMAGE
31	310	KSAM
32	320	USER LOGGING
33	330	DEBUG BREAKPOINT TABLE
34	340	PCB
35	350	SUB-QUEUE MAPPING TABLE
36	360	CILOG
37	25	FILE INTEGRITY
38	380	RIN
39	390	TAPE LABELS
40	87	DEVICE CLASS TABLE
41	400	Reserved
42	401	Cold Load SIR
43		1st JOB
44		2nd JOB
.	.	.
.	.	.

Sir's Ordered by Ranking

RANK	SIR #	SIR NAME
5	16	FMAVT
10	1	LOAD PROCESS
22	17	LOADER SEGMENT TABLE
25	37	FILE INTEGRITY
27	15	JMAT
50	5	PROCESS TREE STRUCTURE
60	6	SCHEDULING QUEUE
70	7	CST ENTRIES
80	8	SYSTEM DIRECTORY
83	27	PV MVTAB
85	10	LDT
87	40	DEVICE CLASS TABLE
90	9	LPDT
91	3	IDD
92	4	ODD
110	11	STORAGE IN OVERLAY AREA
130	13	JPCNT
140	14	JCUT
180	18	VDD
190	19	SPOOK
200	20	MESSAGE CATALOG
210	21	RIT
220	22	VOLUME TABLE
230	23	WELCOME MESSAGE
240	24	ASSOCIATION TABLE
250	25	CS ALLOCATE
260	26	LOGGING BUFFER
280	28	MEASSIR
290	29	PV USER TABLE
300	30	IMAGE
310	31	KSAM
320	32	USER LOGGING
330	33	DEBUG BREAKPOINT TABLE
335	2	CACHE CONTROL
340	34	PCB
350	35	SUB-QUEUE MAPPING TABLE
360	36	CILOG
380	38	RIN
390	39	TAPE LABELS
400	41	Reserved

SIR Table Information

The system internal resource table is located in non-linked memory (resident table). The SIR table is used to protect critical system elements against access by more than one process, i.e., it provides a "lock out" mechanism. Each critical system resource (usually a table) is assigned a specific SIR number. Procedures are provided within MPE to lock (GETSIR) and unlock (RELSIR) the SIR. Processes attempting to obtain a SIR that is not available are impeded by the system. The SIR table entries form the head of a linked list in this case. If more than one process becomes impeded, word 15 of the PCB entry is used to add the "new" process to the growing list. The method of unimpeding the process depends on the SIR type.

A SIR does not respect process priority and operates in a FIFO manner. As processes become impeded on behalf of a SIR the new entries are entered at the tail of the impeded list. When the current holder of the SIR releases it, on the first process in the list (pointed at by the head pointer) is unimpeded. The linked list head and all pointers are then updated and the newly unimpeded process will obtain the SIR.

SIR Entry Formats

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15

|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|                       0                       |0       free
|-----------------------------------------------|     ------------
|                       0                       |1    (not locked)
|-----------------------------------------------|
|                       0                       |2
|-----------------------------------------------|
|                       0                       |3
|-----------------------------------------------|

|-----------------------------------------------|
|              PCB index of holder              |0     SIR locked
|-----------------------------------------------|   ------------------
|                       0                       |1 (no impeded processes
|-----------------------------------------------|
|                       0                       |2
|-----------------------------------------------|
|                       0                       |3
|-----------------------------------------------|

|-----------------------------------------------|
|              PCB index of holder              |0     SIR locked
|-----------------------------------------------|    ----------------
|           SIR QUEUE LENGTH                    |1   (impeded processes)
|-----------------------------------------------|
|         HEAD OF IMPEDED LIST(PCB relative)    |2
|-----------------------------------------------|
|         TAIL OF IMPEDED LIST(PCB relative)    |3
|-----------------------------------------------|
```

P = PIN#
PIN = PCB table entry number
SIR QUEUE LENGTH- number of processes queued for this SIR

The SIR table is indexed by SIR#, with each SIR# corresponding to a unique, pre-assigned system internal resource. Entry #0 is not used. Impeded lists are established by using the SIR table entry (2) as the head of the list and PCB(15) for elements. PINs are always used as pointers, with 0 indicating end of list.

CHAPTER 6 FILE SYSTEM

This chapter describes the MPE V file system. The second section describes the basic concepts. The third section describes the table structures used.

File System Overview

I/O to files is done by reference to file numbers, which are assigned by calling the FOPEN intrinsic. This establishes an initial "point of attachment", which may be described as a connection between a program (i.e., process) and that particular point in a particular file at which the next FREAD or FWRITE would cause data to be transferred. A point of attachment is described by a control block, of which there are several different kinds (described later). Control blocks may exist in the process's own stack or in an extra data segment assigned by the file system. In order to find control blocks quickly, a pointer scheme called vectors is used. A control block is uniquely described by a vector, which consists of two words with the first word containing a segment number and the second word containing a word offset into the control table of the vector table entry which describes the location of the control block within that segment. The entire assemblage, consisting of eight overhead words, the vector table, and all of the control blocks to which it points, comprises the entire segment; if in a stack, it occupies part of the PXFILE part of the PCBX.

The point of attachment is described by a "physical access control block", or PACB, which will exist as a result of an FOPEN to any file (except $NULL). Any required I/O buffers are associated with the PACB; refer to Section 2.1.

All FOPENs specifying "multi-access" for all processes running under a single job use a single PACB for references to a multi-access file. Although all these are attached to a single point in the file, the type of attachment (i.e., AOPTIONS) may be different. So, each FOPEN specifying a multi-access file establishes a "logical access control block", or LACB, which contains the point-of-attachment local values. The use of a single buffer (i.e., PACB) ensures that references by various processes or against various FOPENs within one process are dealt with in strict sequential order. Note that references to a file by other jobs, or by other processes not specifying multi-access, will be through other PACBs, whose buffers will be read or written at the pleasure of the file system; in order to ensure any sort of coherence to such shared references, the jobs must use global RINS and FLOCK and FUNLOCK the file. $STDIN, $STDLIST, and spoolfiles are opened multi-access automatically.

In the case of disc files, there is another kind of control block: the file control block (FCB). It contains copies of information read from the file label, such as the end-of-file pointer, the extent map, and the record and block structure. The EOF pointer is updated in the FCB as the file is written, and all changes made to the FCB are posted to the file label when the file is closed. An FCB is shared by all jobs in the system which reference the file.

The file number assigned by an FOPEN is an index into the Available File Table (AFT), a table of six-word entries which is at the end of the PXFILE part of the PCBX. Two double words are vectors to the PACB and (if it exists) the LACB.

AFT entries can also reside in a global AFT extra data segment. If the file was opened Global AFT (specified in the AOPTIONS) and the program is privileged, then the AFT is placed into this global AFT DST. Any accesses to the file are identical to local AFT's. All accesses to the file opened global must be done from privilege mode code. The file system intrinsics distinguish this file by a negative file number. Again, these files are identical in every other way except for where the AFT entry resides.

Because control blocks are shared among processes, it is necessary to have a scheme for coordinating access to them. A control block is "locked" by a process which requires exclusive access to it for a time. Other processes which attempt to lock the block will find it already locked, and will be impeded and queued. It may also be necessary to lock an entire control block table so that a process can create or destroy a control block in it, or lock or unlock an existing control block in the table.

Another table used by FOPEN is the File Multi-Access Vector Table (FMAVT). This table exists in a system extra data segment and is used by all jobs and processes in the system. When a file is being FOPENed with multi-access specified, the FMAVT is searched; if the file is already open, the FMAVT gives the PACB vector for the prior reference for each job.

Buffers

A bit in AOPTIONS specifies, when a file is opened, whether access is to be buffered or unbuffered. If unbuffered, data is transferred directly between the I/O device and the user's buffer (usually in his stack), which will be frozen in memory for the duration of the transfer. If buffered, the data is moved between the user's buffer and a file system buffer to which the I/O is actually done.

Buffers are associated with the PACB, attached to it as an appendage.

Table Formats

This section gives a detailed discussion of the main tables constructed and used by the file system. The location and overall structure of each table is given, in addition to the table format and a discussion of each field in the table. Table indices at the right of the table are in octal. Index names apply to the entire word; if in parentheses, the names are defined in the file system listing but not explicitly used there.

File System Section of PCBX (PXFILE)

The PXFILE area is a subsection of the PCBX. It is a contiguous, expandable and contractible block of storage that is managed by the file system primarily for its own use. Other subsystems, namely CS and DS, also make use of the PXFILE section. In doing so they must conform to the conventions of the file system.

The overall structure of the PXFILE area is:

OVERHEAD	(FIXED)
CONTROL BLOCK TABLE	(VARIABLE)
AVAILABLE	(VARIABLE)
ACTIVE FILE TABLE	(VARIABLE) DL-5

Overhead

The part labeled Overhead contains information that pertains to the entire section. It is addressed via the pointer at DL-3.

0 1 7	8 15		
PXFILE SIZE IN WORDS		0	PXFSIZE
LAST DOPEN ERROR NO.	LAST COPEN ERROR NO.	1	
N |		2	
LAST DE AFT		3	
SLAVE AFT NUMBER		4	
LAST KOPEN ERROR NUMBER	LAST FOPEN ERROR NUMBER	5	
AFT SIZE IN WORDS		6	PXAFTSIZE
CS TRACE FILE INFO		7 8	(PXCTRINFO)
LAST RESPONDING NO-WAIT I/O AFT ENTRY NUMBER		9	PXFLEFTOFF
1ST USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER		10	PXFCBT1
2ND USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER		11	(PXFCBT2)
3RD USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER		12	(PXFCBT3)
4TH USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER		13	(PXFCBT4)
5TH USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER		14	(PXFCBT5)
6TH USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER		15	(PXFCBT6)
7TH USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER		16	(PXFCBT7)
8TH USER (NOBUF) CONTROL BLOCK TABLE DST NUMBER		17	(PXFCBT8)

Partial word field identifiers are:

PXFDOPEN	= PXFILE(1).(0:8)#,	last DOPEN error code
PXFCOPEN	= PXFILE(1).(8:8)#,	last COPEN error code
PXFNOCB	= PXFILE(2).(0:1)#,	no CB's in PXFILE CBT?
PXFKOPEN	= PXFILE(5).(0:8)#,	last KOPEN error code
PXFFOPEN	= PXFILE(5).(8:8)#,	last FOPEN error code

Discussion:

PXFAFTSIZE This is the size (in words) of the Active File Table (AFT). The size is in words to simplify calculating the size of the available block.

PXFCBT1-8 These are the DST numbers of the user (NOBUF) control block tables. A DST number of 0 indicates that no data segment is allocated.

PXFCOPEN This contains the last COPEN error number. Not used by the file system.

PXFCTRINFO This contains information pertinent to the CS trace file. Not used by the file system.

PXFDOPEN This contains the last DOPEN error number. Not used by the file system.

PXFDSINFO Reserved for DS. Not used by the file system.

PXFFOPEN This contains the last FOPEN error number. If it is zero then the last FOPEN successfully completed; otherwise the last FOPEN was unsuccessful and the number is the file system error number.

PXFKOPEN This contains the last KOPEN error number. KSAM is partly embedded in the file system, and an FOPEN failure on a KSAM file can be caused by a failure to open either the key file or the data file. This error number is used in conjunction with PXFFOPEN to determine which file caused the KSAM open failure. This error number is not used by the file system.

PXFLEFTOFF This is the AFT entry number of the last file/line that completed a nowait I/O; if zero then no nowait I/O has been completed. This cell is maintained solely by and for the IOWAIT intrinsic.

PXFNOCB This bit signifies that control blocks are not to be created in the PXFILE control block table. This bit is set by the NOCB parameter to the CREATE intrinsic or the :RUN command. This feature permits the user to have as much stack space as possible; otherwise the file system will take several hundred words of stack for the PXFILE control block table.

PXFSIZE This is the size (in words) of the complete PXFILE area. It is the sum of the overhead block, the control block table, the active file table and the available block.

PXFILE Control Block Table (PXFCBT)

Addressing within a PXFILE control block table is somewhat more complicated than addressing an extra data segment CBT since the table does not begin at DB+0. As a result all pointers within the table are table relative; the starting address of the table must be added to a pointer to generate a final DB-relative address. This addressing convention is consistently applied to all control block tables.

When the control block table is expanded, space is taken from the AVAILABLE area. If no space is available then the PXFILE area is expanded and the acquired space is added to the AVAILABLE area.

Available Block

The part labeled Available is used to provide space when the Control Block Table or the Active File Table is expanded. These two tables grow towards each other, and when more space is needed it is simply taken from the Available Block.

When the Available area is exhausted, the PXFILE area is expanded, the AFT is relocated and the new space is added to the Available Block.

Currently the PXFILE area is only expanded; it is never contracted.

Active File Table (AFT)

The part labeled Active File Table contains information used by the file system (or CS, DS, etc.) to grossly characterize the file access and, most importantly, to give the location of the control blocks.

The overall structure of the AFT is:

```
 ------------------------
 |                      |
 |       ENTRY N        |     (FIXED, 6 WORDS)
 |                      |
 |----------------------|
 |                      |
 |          .           |
 |          .           |
 |          .           |
 |                      |
 |----------------------|
 |                      |  DL-9
 |       ENTRY 1        |     (FIXED)
 |                      |  DL-5
 ------------------------
```

where N = PXFAFTSIZE/6.

The length of the AFT is specified by PXFAFTSIZE. Unused entries are all zeros. When the table is full it is expanded by taking space from the Available block.

The AFT is negatively indexed by file number: the entry at DL-9 corresponds to file number 1, the entry at DL-15 corresponds to file number 2, etc.

The structure of the global AFT DST, described in Section 2 is as follows:

```
 ------------------------
 |                      | DB + 0
 |  ENTRY 0, NOT USED   |
 |                      |
 |----------------------|
 |                      | DB + 6
 |       ENTRY 1        |
 |                      |
 |----------------------|

 |----------------------|
 |                      | DB+(N*6)
 |       ENTRY N        |
 |                      |
 |----------------------|
```

The structure of a file system AFT entry is:

```
   0  1  2  3  4  5                          15
 ----------------------------------------------
 | ENTRY TYPE  | N |                          |  0
 |--------------------------------------------|
 |          PHYSICAL ACB DST NUMBER           |  1  AFTPACBDST
 |--------------------------------------------|
 |          PHYSICAL ACB ENTRY ADDRESS        |  2  AFTPACBENTRY
 |--------------------------------------------|
 |            LOGICAL ACB DST NUMBER          |  3  AFTLACBDST
 |--------------------------------------------|
 |          LOGICAL ACB ENTRY ADDRESS         |  4  AFTLACBENTRY
 |--------------------------------------------|
 |               NO-WAIT I/O IOQX             |  5  AFTIOQX
 ----------------------------------------------
```

The entry format depends on the entry type; the file system uses entry type 0.

The following partial word field identifiers are used:

```
AFTTYPE      = AFT.(0:4)#,     entry type
AFTNULL      = AFT.(4:1)#,     $NULL file
```

Discussion:

AFTIOQX This is the IOQ index of the pending nowait I/O (if any). This is applicable if the file was opened with the NOWAIT option specified. Also, CS and DS have the same capability and use this cell in a consistent manner. This is because the IOWAIT intrinsic services the file system as well as CS and DS, and is the principal user of this cell. If the IOQX is negative, then one of two possibilities exist. If the file is a message file, then file IOQX is the accessor's reply port. If the file is a standard MPE file, then a read was done to a nonexistent extent and this is simply a stub inserted by the file system.

AFTLACBDST This is the DST that the Logical ACB (LACB) if it exists. This is applicable if the file was opened with the multi-access option specified.

AFTLACBENTRY This is the word offset into the control block table of the LACB vector table entry, applicable if the file was opened with the multi-access option specified.

AFTNULL This bit signifies that the file is $NULL and that there are no control blocks.

AFTPACBDST This is the DST that contains the Physical ACB (PACB). A PACB exists for all files except $NULL.

AFTPACBENTRY This is the word offset into the control block table of the PACB vector table entry. This will be nonzero for all files except $NULL.

AFTTYPE This is the AFT entry type number. At present the following entry types are defined:

0 - file system
1 - remote file
2 - DS (nowait I/O disallowed)
3 - DS (nowait I/O allowed)
4 - CS
5 - CS
6 - KSAM
8 - Message File

Remote file AFT entry:

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| FSTYPE    |            UNUSED              |MR| 0
|-----------------------------------------------|
|                 LINE NUMBER                   | 1
|-----------------------------------------------|
|              REMOTE FILE NUMBER               | 2
|-----------------------------------------------|
|     PENDING FCLOSE DISPOSITION FROM FOPEN     | 3
|-----------------------------------------------|
|                    UNUSED                     | 4
|-----------------------------------------------|
|                     IOQX                      | 5
|-----------------------------------------------|
```

AFT 0
FSTYPE - This value will be 1 for remote files.
MR - Set if the file was opened multi-access.
AFT 1 - Local line number of remote file.
AFT 2 - File number of the remote file.
AFT 3 - Pending disposition of the file. Set when file was FOPEN'd and will possibly be used as the FCLOSE disposition.
AFT 5 - No wait I/O Queue Index.

DS AFT entry:

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|  FSTYPE   | C| M| P| R|  DS ERROR NUMBER      | 0
|-----------------------------------------------|
|            DATA SEGMENT NUMBER                | 1
|-----------------------------------------------|
| DSDCB INDEX    |       UNUSED                 | 2
|-----------------------------------------------|
|               LDEV NUMBER                     | 3
|-----------------------------------------------|
|           PREVIOUS AFT POINTER                | 4
|-----------------------------------------------|
|                 IOQX                          | 5
|-----------------------------------------------|
```

AFT 0
FSTYPE - This field will have the value 2 or 3.
C - On if DSOPEN called by CXDSLINE or REMOTE'HELLO.
M - On if Master PTOP AFT.
P - On if PTOP related.
R - On if remote main process.
AFT 1 - DS data segment table pointer.
AFT 2 - DSDSCB Index - DS data segment control block index.
AFT 3 - Logical device number.
AFT 4 - Preceding DS open AFT Pointer.
AFT 5 - IOQX - Same as described above.

CS Line entry:

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| FYPE     | U| W|ID| B|       UNUSED           | 0
|-----------------------------------------------|
|           LOGICAL DEVICE NUMBER               | 1
|-----------------------------------------------|
|      VECTOR TO MULTIPLE IOQ INDICES           | 2
|-----------------------------------------------|
| TR  | I| R| DIAL|          UNUSED             | 3
|-----------------------------------------------|
|                MISC'DST                       | 4
|-----------------------------------------------|
|             IOQX ( CIO only )                 | 5
|-----------------------------------------------|
```

AFT 0
FTYPE - This value will be 4 or 5. A 5 signifies that the line has an autodialer attached.
W - The line has been opened with no waiting on I/O requests.
ID - Line is a multipoint control or 3270 station.
B - Line was opened with buffering.
AFT 1 - Logical device number of the line.
AFT 2 - Vector to Multiple IOQ indices.
AFT 3
TR - Bit 0 on signifies tracing enabled. Bit 1 on signifies trace all.
I - On if line is currently connected.
R - Signifies that this CS device is an SCCP device.
DIAL - 0 = Dial on write, answer on read.
1 = Answer on write, dial on read.
2 = Always dial.
3 = Never dial.
AFT 4 - DST number of the line's misc data segment.
AFT 5 - If <> 0, then it is the system DB address of a single request IOQ entry. IOWAIT uses this word to pass the IOQ index of the completed request for this AFT to CSIOWAIT.

File Control Block Table (CBTAB)

A file control block table can be located in two places: (a) as a subpart of the PXFILE area, as discussed in Section 3.1.2; or (b) in a data segment. Although putting control block tables in PXFILE has the advantage of providing rapid access, it detracts from the space for the user's stack; so the larger control blocks (or optionally, all control blocks) are put into extra data segments. On the other hand, referencing extra data segments may result in an absence trap, which is slow. Extra data segment control block tables are of three kinds: expandable, nonexpandable, and shared FCB. Nonexpandable CBT's are used for a single PACB with buffers, i.e., where the control block is large or where the control block can't be local to a single process (for multi-access). Expandable (or NOBUF) CBT's are used for small control blocks ,as LACB's, PACB's with no buffers, and FCB's which are local to a single process. A list of the expandable CBT's associated with a process is kept in the overhead area of PXFILE (cf. Section 3.1.1). When a small control block is needed, these CBT's are checked in order to see if one of them has room. Shared FCB CBT's are similar to expandable CBT's except that they belong to the system rather than to a single process; the system keeps a list of DST's which it has assigned for this purpose.

The overall structure of a control block table is:

```
 ------------------------
|                        |
|       OVERHEAD         |     (FIXED, 8 WORDS)
|                        |
|------------------------|
|                        |
|                        |
|     VECTOR TABLE       |     (VARIABLE)
|                        |
|                        |
|------------------------|
|                        |
|                        |
|                        |
|                        |
|     CONTROL BLOCK      |     (VARIABLE)
|         AREA           |
|                        |
|                        |
|                        |
 ------------------------
```

Overhead

The part labeled Overhead contains information pertaining to the entire table.

```
 0   1   2       6    7                          15
---------------------------------------------------
|               TABLE SIZE IN WORDS               |  0  CBTSIZE
|-------------------------------------------------|
|           DST NUMBER CONTAINING TABLE           |  1  CBTDSTX
|-------------------------------------------------|
| TYPE  |      VECTOR TABLE SIZE IN WORDS         |  2
|-------------------------------------------------|
|                    LOCK PIN                     |  3  CBTPIN
|-------------------------------------------------|
| L |                                             |  4  CBTCONTROL
|-------------------------------------------------|
|               IMPEDED QUEUE HEAD                |  5  (CBTQUEUE)
|-------------------------------------------------|
|               IMPEDED QUEUE TAIL                |  6
|-------------------------------------------------|
|                     UNUSED                      |  7
|-------------------------------------------------|
```

Other identifiers used:

```
CBTTYPE    = CBTAB(2).(0:2)    Control block table type
CBTVTSIZE  = CBTAB(2).(2:14)   Vector table size
CBTLOCKBIT= CBTCONTROL.(0:1)  Lock bit
```

Discussion:

CBTDSTX — This is the DST number of the data segment that contains the control block table. If the table is contained in a stack, i.e. in the PXFILE area, then this is the DST number of the stack and not 0.

CBTLOCKBIT — If the entire control block table is locked, then this bit is set. No locking count is kept since control blocks are locked only once from FCREATECB and FDELETECB when control blocks are added to and deleted from the table. The procedure LOCK'CB does not lock the control block because it runs PSEUDODISABLED during the critical times.

CBTQUEUE — This is the impeded queue for the table and has the same format as the impeded queue for a control block in the table. There is no second impeded queue because that facility is used exclusively for BREAK requests against the PACB for $STDIN/$STDLIST.

CBTPIN — This is the PIN number of the process that has the control block locked.

CBTSIZE — This is the size in words of the table. It is initialized when the table is created and changed when the table is expanded. At present a table is never contracted, even though this is possible.

CBTTYPE — This field is the type of the control block table. Possible values are:

0 - stack [PXFILE]
1 - NOBUF (expandable)
2 - System shared FCB
3 - Buffered (Contains a single PACB)

CBTVTSIZE — This is the size, in words, of the vector table area in the control block table. It does not reflect the number of entries used or unused.

NOTE: All PIN's are kept as the word offset into the PCB table and as the actual PIN number.

Vector Table

The part labeled Vector Table contains information used to locate and lock or unlock control blocks in the control block table.

The overall structure of the vector table is:

```
      ------------------------
      |                      |
      |       ENTRY 0        |     (FIXED, 8 WORDS)
      |                      |
      |----------------------|
      |                      |
      |          .           |
      |          .           |
      |          .           |
      |                      |
      |----------------------|
      |                      |
      |       ENTRY N        |     (FIXED)
      |                      |
      ------------------------
```

where N = (CBTVTSIZE/8)-1.

An unused vector table entry will have zeros in all the words of the entry. A used vector table entry will have a nonzero value in the first word of the entry (the control block address is necessarily nonzero).

The general structure of a vector table entry is:

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 |            CONTROL BLOCK ADDRESS                |  0  VT'ADR
 |-------------------------------------------------|
 | L| B|      COUNT      |        UNUSED           |  1  VT'CONTROL
 |-------------------------------------------------|
 |                  LOCK PIN                       |  2  VT'PIN
 |-------------------------------------------------|
 |           HIGH PRIORITY HEAD PIN                |  3  VT'QHEAD
 |-------------------------------------------------|
 |           HIGH PRIORITY TAIL PIN                |  4  VT'QTAIL
 |-------------------------------------------------|
 |           LOW PRIORITY HEAD PIN                 |  5  VT'SAVEDHEAD
 |-------------------------------------------------|
 |           LOW PRIORITY TAIL PIN                 |  6  VT'SAVEDTAIL
 |-------------------------------------------------|
 |                   UNUSED                        |  7
 |-------------------------------------------------|
```

The following partial word identifiers are used:

```
VT'LOCK'BIT  = VT'CONTROL.(0:1)
VT'BREAK'BIT = VT'CONTROL.(1:1)
VT'COUNT     = VT'CONTROL.(2:6)
```

Discussion:

VT'ADR — Control block address is the table relative address of the control block associated with the vector table entry. It is a word displacement from the beginning of the control block table.

VT'BREAK'BIT — This bit signifies that we are in the middle of break mode. This is used for the PACB of $STDIN/$STDLIST from a terminal session only.

VT'LOCK'BIT — This bit is set whenever the control block is locked.

VT'COUNT — This is the count of the number of times that the control block has been locked by the process identified in VT'PIN. If it is zero, then the control block is not locked.

VT'PIN — Contains the PIN of the process which has exclusive access to the control block. Other processes attempting to access the block will be impeded and queued.

VT'QUEUE — The high priority impeded queue is a double word of PINs that are the head and tail of the impeded queue of processes waiting for access to the control block. Processes are impeded and unimpeded by the file system using the normal mechanisms available under MPE.

VT'SAVEDQUEUE — The low priority impeded queue is a double word of PINs and has the same format as VTQUEUE. The only time this word is used is when the control block is in BREAK mode, which can only happen to an ACB corresponding to $STDIN/$STDLIST. It is used to save the current VT'QUEUE when the control block goes into BREAK mode and to restore VT'QUEUE when the control block goes back into non-BREAK mode.

NOTE: All PIN's are stored as offsets within the PCB table and not as actual PIN numbers.

Control Block Area

The part labeled CONTROL BLOCK AREA contains the control blocks used by the file system.

To facilitate storage management, all control blocks have the same overall structure:

```
  0   1   2                                     15
 --------------------------------------------------
 | TYPE  |                  SIZE                  |  0  CBDESCRIP
 |------------------------------------------------|
 |                                                |  1
 |                                                |
 |                                                |
 |                     DATA                       |
 |                                                |
 |                                                |
 |                                                |  N
 --------------------------------------------------
```

where N = Size-1.

Partial word field identifiers are:

```
     CBTYPE          = CB.(0:2)#,      control block type number.
     CBSIZE          = CB.(2:14)#;     control block size
```

Discussion:

CBDESCRIP This is the first word of a control block; the format is common for all control blocks.

CBSIZE This is the size (in words) of the control block. The size includes the descriptor word.

CBTYPE This is the type number of the control block. There are four types of control blocks:

0 - Garbage 1 - FCB 2 - PACB 3 - LACB

When a control block table is created the initial control block area is completely allocated to a single control block of type garbage. When space is requested for a new control block the control block area is scanned (using a first fit algorithm) for a garbage control block that is as large as the size requested. The space for the new control block is taken from this garbage control block and the space remaining becomes the new garbage control block size.

When space is returned it becomes a new garbage control block. To reduce fragmentation the new garbage control block is combined with either of the two neighboring control blocks if they are of type garbage.

If space is requested and no garbage control block is large enough to contain the new control block then the control block area and control block table are expanded by a sufficient amount. If expansion is not possible, some other control block table must be used.

Access Control Block (ACB)

Virtually every file system intrinsic constructs an ACB as its first action. When using the multi-access option, each accessor shares a single PACB. However each accessor is permitted to view the shared file in a slightly different manner than the other accessors. For example, one accessor may access the file in a read-only mode while the other accessors may access the file in a read-write mode. To do this, each accessor must, during his access, have a slightly different ACB.

The PACB holds information that is global to all accessors of the file. The LACB holds information that is local to each accessor of the file. At the beginning of a particular access, an ACB is constructed by calling LOC'ACB, which copies information from both the LACB and the PACB. At the end of the access, the ACB is released by calling UNLOC'ACB; this updates the PACB and LACB from the ACB since some of the fields may have been modified due to the access. This scheme nearly eliminates EXCHANGEDB's to access the various data segments.

Logical Access Control Block (LACB)

All LACBs have the same structure:

```
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
 ----------------------------------------------------------------
 |  3   |                 COMPLETE LACB SIZE                    |  0
 |--------------------------------------------------------------|
 |                                          FILE NUMBER         |  1
 |--------------------------------------------------------------|
 |    FILE NAME - 1ST CHAR.     |     FILE NAME - 2ND CHAR.     |  2
 |--------------------------------------------------------------|
 |    FILE NAME - 3RD CHAR.     |     FILE NAME - 4TH CHAR.     |  3
 |--------------------------------------------------------------|
 |    FILE NAME - 5TH CHAR.     |     FILE NAME - 6TH CHAR.     |  4
 |--------------------------------------------------------------|
 |    FILE NAME - 7TH CHAR.     |     FILE NAME - 8TH CHAR.     |  5
 |--------------------------------------------------------------|
 |                           FOPTIONS                           |  6
 |--------------------------------------------------------------|
 |                           AOPTIONS                           |  7
 |--------------------------------------------------------------|
 |                      RECORD SIZE IN BYTES                    | 10
 |--------------------------------------------------------------|
 |                      BLOCK SIZE IN WORDS                     | 11
 |--------------------------------------------------------------|
 |                             SPARE                            | 12
 |--------------------------------------------------------------|
 |                     CARRIAGE CONTROL CODE                    | 13
 |--------------------------------------------------------------|
 |  |EOF|PG |LN |ST |FK |TC |TB |8B |CAR|DB | EOF T | EOF M |  | 14
 |--------------------------------------------------------------|
 | C|             | TE| IC| Q |   |   TERMINAL STOP CHARACTER   | 15
 |--------------------------------------------------------------|
 |                          ERROR CODE                          | 16
 |--------------------------------------------------------------|
 |                   LAST I/O TRANSMISSION LOG                  | 17
 +--------------------------------------------------------------+
```

Partial word field identifiers are:

```
     LACBSIZE        = LACB.(2:14)#,   size in words
     LACBSTOPCHAR    = LACB(2).(0:8)#, terminal stop character
```

Discussion:

LACBAOPTIONS See ACBAOPTIONS.

LACBBSIZE See ACBBSIZE.

LACBCTL See ACBCTL.

LACBERROR See ACBERROR.

LACBFNUM See ACBFNUM.

LACBFOPTIONS See ACBFOPTIONS.

LACBMODE See ACBMODE.

LACBNAME1-8 See ACBNAME.

LACBPACB This is the DST and vector table entry for the Physical ACB (PACB) for the file. file.

LACBRSIZE See ACBRSIZE.

LACBSIZE This is the size, in words, of the LACB. All LACBs are eighteen (decimal) words long.

LACBSTATE See ACBLSTATE.

LACBSTOPCHAR See ACBSTOPCHAR.

LACBTLOG See ACBTLOG.

Physical Access Control Block (PACB)

The overall structure of the PACB is:

```
    -------------------------
    |                       |
    |      BASIC PACB       |      (FIXED)
    |                       |
    |-----------------------|
    |                       |
    |       BUFFERING       |
    |                       |      (VARIABLE)
    |       EXTENSION       |
    |                       |
    -------------------------
```

The buffering extension is optional; it is present if and only if the file is accessed with buffering. There are thus two possible formats for an ACB:

1. No buffers; the buffering extension is not present.

2. PACB buffers; the buffering extension is present and the buffers are in the buffering extension.

If multiple PACB buffers exist, there will be a buffering extension for each, immediately preceding the buffer. The basic PACB (or NOBUF PACB) is copied into the the ACB as words 0 through %63; an ACB "extension" is then generated in words %64 - %67. The resulting ACB thus has the following format:

```
      0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
    -----------------------------------------------------
 0  | 2  |                COMPLETE ACB SIZE              |  0
    |-----------------------------------------------------|
 1  |                        |         FILE NUMBER       |  1
    |-----------------------------------------------------|
 2  |  FILE NAME - 1ST CHAR. |  FILE NAME - 2ND CHAR.    |  2
    |-----------------------------------------------------|
 3  |  FILE NAME - 3RD CHAR. |  FILE NAME - 4TH CHAR.    |  3
    |-----------------------------------------------------|
 4  |  FILE NAME - 5TH CHAR. |  FILE NAME - 6TH CHAR.    |  4
    |-----------------------------------------------------|
 5  |  FILE NAME - 7TH CHAR. |  FILE NAME - 8TH CHAR.    |  5
    |-----------------------------------------------------|
 6  |                     FOPTIONS                        |  6
    |-----------------------------------------------------|
 7  |                     AOPTIONS                        |  7
    |-----------------------------------------------------|
 8  |                Record size in bytes                 |  10
    |-----------------------------------------------------|
 9  |                BLOCK SIZE IN WORDS                  |  11
    |-----------------------------------------------------|
 10 |                      UNUSED                         |  12
    |-----------------------------------------------------|
 11 |               CARRIAGE CONTROL CODE                 |  13
    |-----------------------------------------------------|
 12 |  |EOF|PG |LN |ST |FK |TC |TB |8B |CAR|DB | EOF T | EOF M |  14
    |-----------------------------------------------------|
 13 | C|           | TE| IC| Q |   |  TERMINAL STOP CHARACTER |  15
    |-----------------------------------------------------|
 14 |                    ERROR CODE                       |  16
    |-----------------------------------------------------|
 15 |             LAST I/O TRANSMISSION LOG               |  17
    +-----------------------------------------------------+
 16 |                                                     |  20
    |                   FILE POINTER                      |
 17 |                                                     |  21
    |-----------------------------------------------------|
 18 |                                                     |  22
    |           CURRENT VARIABLE BLOCK NUMBER             |
 19 |                                                     |  23
    |-----------------------------------------------------|
 20 |                                                     |  24
    |              RECORD TRANSFER COUNT                  |
 21 |                                                     |  25
    |-----------------------------------------------------|
 22 |                                                     |  26
    |               BLOCK TRANSFER COUNT                  |
 23 |                                                     |  27
    |-----------------------------------------------------|
 24 |                                                     |  30
    |           HIGHEST BLOCK NUMBER STARTED              |
 25 |                                                     |  31
    |-----------------------------------------------------|
```

```
 26 |                                                     |  32
    |                    FCB VECTOR                       |
 27 |                                                     |  33
    |-----------------------------------------------------|
 28 |              TOTAL NUMBER OF LACB'S                 |  34
    |-----------------------------------------------------|
 29 |  |BK |    DEVICE TYPE     |  LAST LOGICAL I/O STATUS  |  35
    |-----------------------------------------------------|
 30 |              LOGICAL DEVICE NUMBER                  |  36
    |-----------------------------------------------------|
 31 |PF |HIT|     | CURRENT BUFFER| TAPE DISPLACE |  NO. BUFFERS |  37
    |-----------------------------------------------------|
 32 |            CURRENT RECORD WORD INDEX                |  40
    |-----------------------------------------------------|
 33 |                   BUFFER SIZE                       |  41
    |-----------------------------------------------------|
 34 |            VIRTUAL LOGICAL DEVICE NO.               |  42
    |-----------------------------------------------------|
 35 |                   FMAVT INDEX                       |  43
    |-----------------------------------------------------|
 36 |              NUMBER OF INPUT LACB'S                 |  44
    |-----------------------------------------------------|
 37 |       NAME TYPE          |    FILE DISPOSITION      |  45
    |-----------------------------------------------------|
 38 |     ACCESS BIT MAP       |    BLOCKING FACTOR       |  46
    |-----------------------------------------------------|
 39 | S | M | Q | R | D |      | AE| RW|ABR| NE| SEOFS| EOFS |  47
    |-----------------------------------------------------|
 40 |  SPOOLED DEVICE TYPE  |  SPOOLED DEVICE RECORD SIZE |  50
    |-----------------------------------------------------|
 41 |             SPOOLED DEVICE FOPTIONS                 |  51
    |-----------------------------------------------------|
 42 |             SPOOLED DEVICE AOPTIONS                 |  52
    |-----------------------------------------------------|
 43 |                IDD OR ODD INDEX                     |  53
    |-----------------------------------------------------|
 44 |                                                     |  54
    |               NO-WAIT DISK ADDRESS                  |
 45 |                                                     |  55
    |-----------------------------------------------------|
 46 |                     UNUSED                          |  56
    |-----------------------------------------------------|
 47 |           NO-WAIT LOGICAL DEVICE                    |  57
    |-----------------------------------------------------|
 48 |                                                     |  60
    |            P1P2 USED BY FDEVICECONTROL              |
 49 |                                                     |  61
    |-----------------------------------------------------|
 50 |                     UNUSED                          |  62
    |-----------------------------------------------------|
 51 |                     UNUSED                          |  63
    +-----------------------------------------------------+
```

The above words, 0-%63, are physically located in the PACB of the file. Below, words %64-%67, are used by file system intrinsics- and are placed onto the stack by the procedure LOC'ACB when locking the ACB. Therefore, the buffering extension, if pres- ent, will immediately follow word %63 of the actual ACB in the Control Block Table of the file.

```
    |-----------------------------------------------------|
 52 |            DST RELATIVE OFFSET TO PACB              |  64
    |-----------------------------------------------------|
 53 |            DST RELATIVE OFFSET TO LACB              |  65
    |-----------------------------------------------------|
 54 |      DST RELATIVE OFFSET TO ACB IN THE STACK        |  66
    |-----------------------------------------------------|
 55 |            STACK RELATIVE OFFSET TO DB              |  67
    |-----------------------------------------------------|
```

The following identifiers are used when referring to an ACB:

```
    (ACBSIZE)     = ACB.(2:14)#,         size in words
    ACBFNUM       = ACB(1).(8:8)#,       file number
    ACBNAME       = ACB(2)#,             file name
    ACBNAME1      = ACBDBL(1)#,          file name - first half
    ACBNAME2      = ACBDBL(2)#,          file name - second half
    ACBFOPTIONS   = ACB(6)#,             FOPTIONS
    ACBAOPTIONS   = ACB(7)#,             AOPTIONS
    ACBRSIZE      = ACB(8)#,             record size (bytes)
    ACBBSIZE      = ACB(9)#,             block size (words)
    Spare         = ACB(10)#,            Unused
    ACBCTL        = ACB(11)#,            carriage control word
    ACBLSTATE     = ACB(12)#,            local state flags
    ACBEOF        = ACBLSTATE.(1:1)#,    end of file sensed
    ACBLPCTL      = ACBLSTATE.(2:2)#,    page and line control
    ACBPAGECTL    = ACBLSTATE.(2:1)#,    page control
    ACBLINECTL    = ACBLSTATE.(3:1)#,    line control
    ACBSTREAM     = ACBLSTATE.(4:1)#,    stream I/O
    ACBFKEYS      = ACBLSTATE.(5:1)#,    restore function keys
    ACBXMITCRLF   = ACBLSTATE.(6:1)#,    transmit CR,LF to user
    ACBTBLOCK     = ACBLSTATE.(7:1)#,    disable block mode
    ACBBINARYIO   = ACBLSTATE.(8:1)#,    8-bit terminal transfers
    ACBCARRIAGE   = ACBLSTATE.(9:1)#,    carriage control flag
    (ACBDEFBLOCK) = ACBLSTATE.(10:1)#,   default blocking
    ACBREADCODE   = ACBLSTATE.(11:4)#,   input EOF check
    ACBREADTYPE   = ACBLSTATE.(11:2)#,   input EOF type
    ACBREADMODE   = ACBLSTATE.(13:2)#;   input EOF mode
    ACBMODW       = ACB(13)#,            mode word
    ACBMODE       = ACBMODW.(0:8)#,      mode setting
    ACBCIROVERFLOW= ACBMODW.(0:1)#,      Signifies CIR overflow
    ACBSETMODE    = ACBMODW.(4:4)#,      FSETMODE bits
    ACBTAPEERROR  = ACBMODW.(4:1)#,      report recovered tape error
    ACBINHIBCRLF  = ACBMODW.(5:1)#,      inhibit terminal CR/LF
    ACBQUIESCE    = ACBMODW.(6:1)#,      critical output verify
    ACBSTOPCHAR   = ACBMODW.(8:8)#,      terminal stop character
```

```
ACBERROR     = ACB(14)#,            error code
ACBTLOG      = ACB(15)#,            last I/O transmission log
ACBFPTR      = ACBDBL(08)#,         current record number
ACBBLK       = ACBDBL(09)#,         current variable block
ACBRTFRCT    = ACBDBL(10)#,         logical record TFR count
ACBBTFRCT    = ACBDBL(11)#,         block transfer count
ACBHIBLK     = ACBDBL(12)#,         highest block started
ACBFCBV      = ACBDBL(13)#,         FCB Vector table entry
ACBSHCNT     = ACB(28)#,            # of LACBs
ACBSTATW     = ACB(29)#,            access class, status, etc.
ACBBREAK     = ACBSTATW.(1:1)#,     break ($STDIN/LIST only)
ACBDTYPE     = ACBSTATW.(2:6)#,     device type
ACBACCCL     = ACBSTATW.(2:3)#,     device access class
ACBSUBCL     = ACBSTATW.(5:3)#,     device sub-class
ACBSTATUS    = ACBSTATW.(8:8)#,     last logical I/O status
ACBQSTATUS   = ACBSTATW.(8:5)#,     qualifying status part
ACBGSTATUS   = ACBSTATW.(13:3)#,    general status part
ACBDADDR     = ACB(30)#,            Ldev number of file
ACBBUFX      = ACB(31)#,            buffer data & misc. flags
ACBPRIV      = ACBBUFX.(0:1)#,      privileged access only
ACBHIT       = ACBBUFX.(1:1)#,      buffer hit flag
ACBCURRBUF   = ACBBUFX.(4:4)#,      current buffer nor.
ACBNUMBUFS   = ACBBUFX.(12:4)#,     number of buffers less 1
ACBBUFUSED   = ACB(32)#,            used block word count
ACBBUFSIZE   = ACB(33)#,            buffer size (words)
ACBSPVDEV    = ACB(34)#,            spooled virtual device
ACBFMAVTX    = ACB(35)#,            FMAVT index
ACBSHCNTIN   = ACB(36)#,            Number of input LACB's
ACBDNTD      = ACB(37)#,            type & disposition
ACBDNTYPE    = ACBDNTD.(0:8)#,      name type for dir. search
ACBDISP      = ACBDNTD.(8:8)#,      file disposition
ACBAMLD      = ACB(38)#,            access mask & LDEV
ACBACCESS    = ACBAMLD.(0:8)#,      access mask
ACBBLKFACT   = ACBAMLD.(8:8)#,      Blocking factor of file
ACBGSTW      = ACB(39)#,            spool control flags
ACBSPOOLED   = ACBGSTW.(0:1)#,      spooled device flag
ACBSPOOLIO   = ACBGSTW.(0:2)#,      spooled IN/OUT
ACBSPSQ      = ACBGSTW.(2:2)#,      squeeze flags
ACBSPSQZ     = ACBGSTW.(2:1)#,      file squeezed
ACBSPRSQ     = ACBGSTW.(3:1)#,      request to squeeze
ACBSPDSQ     = ACBGSTW.(4:1)#,      squeeze just done
ACBNOWAITEOF = ACBGSTW.(8:1)#,      EOF advanced?
ACBNOWAITMODE = ACBGSTW.(9:1)#,     last I/O: 0=read, 1=write
ACBABORTREAD = ACBGSTW.(10:1)#,     abort broken re-read?
ACBNEWEOF    = ACBGSTW.(11:1)#,     EOF advanced - tape file
ACBSAVEEOFS  = ACBGSTW.(12:2)#,     for saving ACBEOFS
ACBEOFS      = ACBGSTW.(14:2)#,     EOF flags - :EOD/:
ACBSPTYRC    = ACB(40)#,            spooled dev type/recsize
ACBSPTYPE    = ACBSPTYRC.(0:6)#,    spooled dev type
ACBSPREC     = ACBSPTYRC.(6:10)#,   spooled dev rec size
ACBSPFOPT    = ACB(41)#,            spooled dev FOPTIONS
ACBSPAOPT    = ACB(42)#,            spooled dev AOPTIONS
ACBSPXDDX    = ACB(43)#,            IDD/ODD index
ACBNOWAITDA  = ACBDBL(22)#,         Nowait disc address
```

```
Spare         = ACB(46)#,          Unused
ACBNOWAITLDEV = ACB(47)#,          Nowait logical device
ACBP1P2       = ACBDBL(24)#,       Used by FDEVICECONTROL
ACBP1         = ACB(48)#,            "   "    "    "
ACBP2         = ACB(49)#;            "   "    "    "
```

Discussion:

ACBABORTREAD   This flag is used to abort a broken terminal re-read. The flag is set via the ABORT parameter to FUNBREAK. If the flag is set then the READ PENDING message will be aborted along with the re-read. This feature is needed to handle the BREAK...:ABORT, etc. situation.

ACBACCCL   This is the access class part of the device type number. The following are legal values:

0 - direct (e.g. disc)
1 - serial input (e.g. card reader)
2 - parallel input/output (e.g. terminal)
3 - serial input/output (e.g. magnetic tape)
4 - serial output (e.g. line printer)

ACBACCESS   This is the access bit map for the file. The following are the bit definitions of this eight-bit field:

(0:1) - unused
(1:1) - unused
(2:1) - read
(3:1) - append
(4:1) - write
(5:1) - lock
(6:1) - execute
(7:1) - save

This access security is determined by the ACCCHECK intrinsic and enforced by the file system.

ACBAOPTIONS   This is the AOPTIONS in effect for this file access.

ACBBINARYIO   This bit controls full eight bit transfers on the 2644 page mode terminal. It is adjusted by FCONTROL(26) and FCONTROL(27).

ACBBLK   This is the block number of the current variable record format block. Applicable if the record format is variable.

ACBBLKFACT   This is the blocking factor for the file. It is the number of records in a block. Legal values range from 1 to 255.

ACBBREAK   This is the break mode flag. It is applicable if the ACB is for $STDIN or $STDLIST. If set it means that the BREAK key has been hit and that the CI should have high priority access to the ACB. The flag will be cleared when a RESUME or ABORT is issued.

ACBBSIZE   This is the block size, in words, of the file.

ACBBTFRCT   This is the total number of blocks transferred to and from the file. The initial value is 0D.

ACBBUFUSED   This is the word index, relative to the base of the block, for the selected record within the block. This is applicable if the file access is buffered.

ACBCARRIAGE   This bit signifies that the file has carriage control. It is the same as the carriage control bit in ACBFOPTIONS if the file is spooled. If not spooled, the bit is zero, and IOMOVE will pass the FWRITE carriage control parameter directly to the driver rather than embedding it as the first character of the output record.

ACBCTL   This is the CONTROL parameter from the last FWRITE. This value is pertinent if the file was opened with carriage control.

ACBCURRBUF   This is the buffer number (0-relative) containing the most recently referenced record. Applicable if the file access is buffered.

ACBDADDR   This is the logical device number of the file. For a disc file this is the logical device number of the first extent.

ACBDEFBLOCK   This bit signifies that the file is to be accessed with default blocking. The bit is initialized from the FOPEN stateword STATE. It does not need to be in the ACB; it is mentioned here only to signify that the bit is effectively used due to the way ACBLSTATE is initialized from STATE.

ACBDISP   This is the file close disposition derived from the FOPEN call. The only way this can be specified is via a file equation. The legal values are the same as those for FCLOSE. ACBDNTYPE   This is the file reference format type number and is derived from the FOPEN call. The following are legal values:

0 - full name
1 - account name absent
2 - group and account name absent
3 - null name

This information is needed by FRENAME.

ACBDTYPE   This is the device type number of the file. The following are legal values (octal):

0 - moving head disc
1 - fixed head disc
7 - foreign disc
10 - card reader
11 - paper tape reader
20 - terminal
24 - card reader/interpreter/punch
26 - SSLC
27 - programmable controller
30 - magnetic tape
31 - serial disc
40 - line printer
41 - card punch
42 - paper tape punch
43 - CALCOMP 500 plotter
44 - CALCOMP 600 plotter
45 - CALCOMP 700 plotter

ACBEOF   This bit is set when EOF has been sensed.

ACBEOFS   This is the type of EOF detected on $STDIN(X). This field consists of two bits:

(0:1) - super colon (i.e. EOF for $STDINX)
(1:1) - regular colon (i.e. EOF for $STDIN)

Applicable for multi-access to $STDIN(X) only.

ACBERROR   This is the error number for the file. It is used by all intrinsics except FOPEN. When an error is detected the error number is placed in this cell. The error number is cleared at the beginning of each callable intrinsic except FCHECK (which reads it).

ACBFCB   This is the FCB vector for the file. Applicable only to disc files.

ACBFKEYS   This bit controls the definition of the f1 and f2 function keys on the 2644 page mode terminal; it is adjusted by FCONTROL(32) and FCONTROL(33). (Obsolete function)

ACBFNUM   File number, range from 1 to 255. Used mostly for calling routines that access things such as labels by file number.

ACBFOPTIONS   This is the FOPTIONS in effect for this file access.

ACBFPTR   This is the sequential access record pointer; it contains the next sequential record number. The initial value is 0D. This value is used only by the FREAD, FWRITE and FUPDATE intrinsics. However the value is maintained by all data transferring file system intrinsics.

ACBFMAVTX
This is the entry index into the file multi-access vector table (FMAVT). This is valid if the file access is multi-access.

ACBGSTATE
These are miscellaneous state flags. These are "global" in nature in that they are the same for all accessors in a multi-access environment. The constituent bits are described individually.

ACBGSTATUS
This is the general part of the last I/O status for the file. The following are the legal values:

0 - pending
1 - successful
2 - end of file
3 - unusual condition
4 - irrecoverable error

ACBHIBLK
This is the highest block number for which an anticipatory read has been issued, and is applicable if the file access is buffered. The initial value is -1D.

ACBHIT
This is the buffer hit flag. If set it indicates that the last read or write request was serviced without any physical I/O required. This flag is used only for performance measurement. The code which manipulates it is optional to the file system, and is controlled by compiler toggle X3.

ACBINHIBCRLF
This bit controls the termination of lines written to the terminal. If not set then each line is terminated with a CR and LF; if set then no line termination characters are used. This bit is valid if the file is a terminal file; it is adjusted by FSETMODE.

ACBLINECTL
This is the line control bit. If not set then each line is post-spaced; if set then each line is prespaced. This bit is used by line printers and terminals only. It is adjusted by FCONTROL(1) and FWRITE with the appropriate carriage control.

ACBLPCTL
This are the line and page control bits, which are described separately.

ACBLSTATE
These are miscellaneous state flags. They are "local" in nature in that they may be different for each accessor in a multi-access environment. Bits (9:6) are initialized from the stateword local variable called STATE in FOPEN; the ten remaining bits are initialized individually. The constituent bits are described individually.

ACBMODE
These are miscellaneous mode flags. The constituent bits are described individually.

ACBNAME
This is the local file name. The name is eight bytes in length with trailing blanks added.

ACBNEWEOF
This flag when set indicates that a new tape mark should be written before the tape is rewound or backspaced. Applicable only to magnetic tape files.

ACBNOWAITEOF
This bit is used to save the value of the local EOF advanced flag NEWEOF in IOMOVE between the I/O initiation and I/O completion calls. This flag is applicable if the file is accessed in nowait I/O mode.

ACBNOWAITMODE
This cell is used to save the I/O mode between nowait I/O initiation and completion calls. If the bit is set then the last I/O request was a write; otherwise it was a read. This cell is pertinent if the file is accessed in nowait I/O mode.

ACBNUMBUFS
This is the number of buffers, less one, used for the file access. Applicable if the file access is buffered.

ACBPAGECTL
This is the page control bit. If not set then a page is assumed to consist of 60 lines (auto page eject); if set then a page is assumed to consist of 66 lines (no auto page eject). This is used primarily for line printers but is also valid for terminals; these are the only devices for which this is valid. This bit is adjusted by FCONTROL(1) and FWRITE with the appropriate carriage control.

ACBPRIV
This flag when set indicates that the file is privileged in that it has a negative file code; the user must be in privileged mode to access it.

ACBQSTATUS
This is the qualifying part of the last I/O status for the file. The values are unique for each general status part. See I/O System IMS for all legal values.

ACBQUIESCE
This bit controls critical output verification. If set, buffered output is guaranteed to have been written to the device when control is returned to the user. This bit is adjusted by FSETMODE.

ACBREADCODE
This field consists of the input EOF checking type and mode, and is used to generate the P1 parameter to ATTACHIO. These fields are described individually.

ACBREADMODE
This field controls the input EOF checking mode. It is 00 for reading $STDIN, 01 for reading $STDINX, and 10 for the command interpreter.

ACBREADTYPE
This field controls the input EOF checking type. It is 01 for JOBs, 10 for SESSIONs, and 00 for DATA.

ACBRSIZE
This is the file's record size in positive bytes.

ACBRTFRCT
This is the total number of records transferred to and from the file. The initial value is 0D.

ACBSAVEEOFS
This field is used to save the contents of ACBEOFS during BREAK mode processing.

ACBSHCNT
This is the total number of LACBs that exist for this PACB. Valid if the file access is multi-access.

ACBSHCNTIN
This is the total number of input-only LACBs that exist for this PACB. Valid if the file access is multi-access.

ACBSHCNTS
This is the total LACB and total input-only LACB counts, each of which is described separately.

ACBSIZE
This is the size, in words, of the ACB. The complete size (including buffers) may be calculated from the DST size containing the ABC. It does not include the buffering extension, if present.

ACBSPAOPT
This is the AOPTIONS for the spooled device. Applicable if the file access is to a spooled device.

ACBSPFOPT
This is the FOPTIONS for the spooled device. Applicable if the file access is to a spooled device.

ACBSPOOLED
This is the spooled device flag. If set then the file access is to a spooled device.

ACBSPOOLIO
This field is a combination of the spooled device flag and the input/output mode of the spooled device. Legal values are:

00 - not spooled
01 - illegal
10 - input spooling
11 - output spooling

ACBSPREC
This is the record size, in bytes, of the spooled device. Applicable if the file access is to a spooled device.

ACBSPTYPE
This is the device type (from the LDT) of the spooled device. Applicable if the file access is to a spooled device.

ACBSPTYRC
This cell contains the spooled device type and record size, which are described separately.

ACBSPVDEV
This is the logical device number of the spooled device. Applicable if the file access is to a spooled device.

ACBSPXDDX
This is the index into the IDD or ODD for a spoolfile. Applicable if the file access is to either a spooled device or a spoolfile.

ACBSTATUS
This is the last I/O status for the file. It comes from the I/O status part of the IOCB returned by ATTACHIO. Not all ATTACHIO calls update this cell.

ACBSTOPCHAR
This is the record termination character used for terminal reads. This character can be changed via FCONTROL(25).

ACBSTREAM
This bit signifies inter-block garbage for disc files. If set, the block size is a multiple of 128 words and therefore there is no garbage data between blocks. This fact is used to improve multirecord I/O by mapping the request into as few ATTACHIOs as possible.

ACBSUBCL
This is the sub-class part of the device type number. The sub-class is unique for each access class. The following are the legal sub-class values for each device class:

0 - direct
0 - moving head disc
1 - fixed head disc
7 - foreign disc
1 - serial input
0 - card reader
1 - paper tape reader
2 - parallel input/output
0 - terminal
4 - card reader/punch
6 - SSLC
7 - programmable controller
3 - serial input/output
0 - magnetic tape
7 - serial disc
4 - serial output
0 - line printer
1 - card punch
2 - paper tape punch
3 - CALCOMP 500 plotter
4 - CALCOMP 600 plotter
5 - CALCOMP 700 plotter

ACBTAPEERROR This bit controls the reporting of recovered magnetic errors. If not set the recovered errors are not reported to the user; if set then recovered errors are reported to the user by returning CCL and error number 39. Valid if the file is a magnetic tape file. This bit is adjusted by FSETMODE.

ACBTBLOCK This bit controls block mode transfers on the 2644 page mode terminal. This bit is adjusted by FCONTROL(28) and FCONTROL(29).

ACBTLOG This is the last I/O transmission log for the file. It comes from the I/O transmission log part of the IOCB returned by ATTACHIO. Not all ATTACHIO calls update this cell.

ACBVDADDR This is the volume table index for the file. Applicable if the file is a disc file.

ACBXMITCRLF This bit controls CR and LF insertion into the user buffer on the 2644 page mode terminal. This bit is adjusted by FCONTROL(30) and FCONTROL(31).

If present, the PACB buffering extension contains from one to sixteen block buffers each having the following format:

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|               IOQ ENTRY INDEX                 |  0  BLKIOQX
|-----------------------------------------------|
| BLK LDEV NUMBER         |  |  | U| R| D| W| M| P|  1  BLKFLAGW
|-----------------------------------------------|
|                 IOCB - STATUS                 |  2  BLKLSTAT
|-----------------------------------------------|
|           IOCB - TRANSMISSION LOG             |  3  BLKTLOG
|-----------------------------------------------|
|                                               |  4  BLKBLOCK
|                 BLOCK NUMBER                  |
|                                               |  5
|-----------------------------------------------|
|                                               |  6  BLKDADDR
|             BLOCK SECTOR ADDRESS              |
|                                               |  7
|-----------------------------------------------|
|                                               |  8  BLKEXTBASE
|              BLOCK EXTENT BASE                |
|                                               |  9
|-----------------------------------------------|
|              BLOCK EXTENT SIZE                |  10 BLKEXTSIZE
|-----------------------------------------------|
|                   UNUSED                      |  11
|-----------------------------------------------|
|                                               |  12 BLKBUFFER
|                                               |
|                                               |
|                   BUFFER                      |
|                                               |
|                                               |
|                                               |
|-----------------------------------------------|
```

Other identifiers used:

```
     BLKFLAGW        = BLK(1)#,          Flag and LDEV word
     BLKLDEV         = BLKFLAGW.(0:8)#, block logical device number
     BLKFLAGS        = BLKFLAGW.(0:8)#, block I/O flags
     BLKUNALLOCEXT   = BLKFLAGW.(10:1), Block from unalloc. extent
     BLKREVERSE      = BLKFLAGW.(11:1), FREADBACKWARD (not used )
     BLKDONTWAIT     = BLKFLAGW.(12:1), I/O status not checked
     BLKIOOUT        = BLKFLAGW.(13:1)#,last I/O was write?
     BLKDIRTY        = BLKFLAGW.(14:1)#,buffer modified?
     BLKIOPEND       = BLKFLAGW.(15:1)#,I/O in progress?
     BLKIOCOMP       = BLKFLAGW.(14:2)#,I/O complete - not dirty
     BLKIOCB         = BLKDBL(1)#,       IOCB
```

Discussion:

BLKBLOCK This is the block number of the data contained in the buffer. A value of -10 indicates that the buffer is empty.

BLKBUFFER This is the actual file system buffer space. Each buffer is exactly one file block in size.

BLKDADDR This is the block's logical device and sector number.

BLKDIRTY This flag is set if the contents of the buffer has been modified. When the block buffer is re-used this flag is checked to see if the block needs to be written to the device.

BLKDONTWAIT This bit will be on if the I/O was already completed via "DONT'WAIT" but the status has not been checked yet. Check the status before using the block in the buffer.

BLKEXTBASE This is the sector address of the extent base in which the block resides. This is used for disc caching.

BLKEXTSIZE The size, in sectors, of the extent in which the block resides. This is used for disc caching.

BLKFLAGS These are the miscellaneous flags associated with the block, which are described separately.

BLKIOCB This is the IOCB returned by the I/O system when the block I/O has completed. On a blocked I/O request this is obtained from the ATTACHIO call; on an unblocked I/O request this is obtained from WAITFORIO.

BLKIOCOMP This is the buffer modified flag (BLKDIRTY) and the I/O in progress flag (BLKIOPEND), which are described separately. This field is usually interrogated to see if it contains the value 2, which means that the buffer has been modified but not yet written to the device.

BLKIOOUT This is the mode of the I/O operation for the block. It is set by a write and cleared by a read.

BLKIOPEND This is the I/O in progress flag. It is set if the I/O is pending; it is cleared when the I/O has completed.

BLKIOQX This is the IOQ index of the unblocked I/O request for the block. It is used as the argument to WAITFORIO, which ensures the completion of the I/O request.

BLKLDEV This is the logical device number of the block. (Valid only for disc files.)

BLKLSTAT The I/O status part of the IOCB consists of the PCB number and the error code for the completed I/O request.

BLKTLOG The transmission log part of the IOCB is the number of words or bytes transferred by the the I/O request.

BLKREVERSE This bit would indicate that we are reading back- wards from a tape. However, currently FREADBACK- WARDS can only be performed unbuffered.

BLKUNALLOCEXT This bit signifies that the block was "read" from an unallocated extent. Actually, the buffer was simply cleared with fill characters. Therefore, if a write is attempted to the block residing in this buffer, it must pass through FCONVBLK to allocate the extent first.

File Control Block (FCB)

The FCB coordinates access to a file on a sharable device. At present the only sharable device is a disc, so only disc files have FCBs.

The information contained in an FCB is derived from the file label. The FCB is used to hold this information, rather than the file label, since it can be accessed more quickly.

There are two strategies to choose from in deciding where to place the FCB. If the file has been opened exclusive and no other process could possible share this file, then the FCB is placed into the PXFILE area (or in a NOBUF expandable CBT if it won't fit in the PXFILE area or if the program is run with NOCB). If the file could possible be shared, then the FCB is always placed in a shared control block table. The number of a data segment containing a list of shared file system data segments is kept in system global location 1076 octal. The size of the FCB depends on the maximum number of extents specified at FOPEN; there are 44 (octal) words plus two per extent. There will be at least one extent, since the file label always exists in the first extent. The FCB extent map is in terms of logical device and sector number. The extent map in the file label is in terms of volume rather than logical device; the map is converted by VTABOLDEV when the label is read, and converted back by LDEVTOVTAB when the label is written to disc.

The FCB has the following format:

```
    0   1   2   3         7   8        12  13  14  15
  ---------------------------------------------------
0 |   1   |              COMPLETE FCB SIZE           |  0
  |--------------------------------------------------|
1 |                      SPARE                       |  1
  |--------------------------------------------------|
2 |                      FOPTIONS                    |  2  FCBFOP-
  |--------------------------------------------------|     TIONS
3 |               DEVICE SPECIFICATION               |  3  FCBDEVICE
  |--------------------------------------------------|
4 | PREV. LOCK|  DEV. TYPE  | C |     |DEVICE SUBTYPE |  4
  |--------------------------------------------------|
5 |               NO. OPENS FOR OUTPUT               |  5
  |--------------------------------------------------|
6 |               NO. OPENS FOR ANY MODE             |  6
  |--------------------------------------------------|
7 |                  RIN NUMBER                      |  7  FCBRIN
  |--------------------------------------------------|
8 |                EXCLUSIVE STATUS                  | 10  FCBEXC-
  |--------------------------------------------------|     STAT
9 | C|         | MVTABX  |        VMASK              | 11  FCBPVINFO
  |--------------------------------------------------|
10|                                                  | 12  FCBFLIM
  |                 FILE LIMIT                       |
11|                                                  | 13
  |--------------------------------------------------|
```

```
12 |                     UNUSED                       | 14
   |--------------------------------------------------|
13 |                     UNUSED                       | 15
   |--------------------------------------------------|
14 |                                                  | 16  FCBEOF
   |              END OF DATA POINTER                 |
15 |                                                  | 17
   |--------------------------------------------------|
16 | NO. USER LABELS WRITTEN |  NO. USER LABELS AVAIL. | 20  FCBUSERLBL
   |--------------------------------------------------|
17 |             EXTENT SIZE IN SECTORS               | 21  FCBEXTSIZE
   |--------------------------------------------------|
18 |     BLOCKING FACTOR      |    SECTORS PER BLOCK  | 22
   |--------------------------------------------------|
19 | SECTOR OFFSET TO DATA    | DISP | NO. EXTENTS - 1| 23
   |--------------------------------------------------|
20 |           LAST EXTENT SIZE IN SECTORS            | 24  FCBLAST-
   |--------------------------------------------------|     EXTSIZE
21 |              NO. OPENS INPUT MODE                | 25
   |--------------------------------------------------|
22 | GROUP NAME - 1ST CHAR.  | GROUP NAME - 2ND CHAR. | 26  FCBGN
   |--------------------------------------------------|
23 | GROUP NAME - 3RD CHAR.  | GROUP NAME - 4TH CHAR. | 27
   |--------------------------------------------------|
24 | GROUP NAME - 5TH CHAR.  | GROUP NAME - 6TH CHAR. | 30
   |--------------------------------------------------|
25 | GROUP NAME - 7TH CHAR.  | GROUP NAME - 8TH CHAR. | 31
   |--------------------------------------------------|
26 |  ACCT NAME - 1ST CHAR.  |  ACCT NAME - 2ND CHAR. | 32  FCBAN
   |--------------------------------------------------|
27 |  ACCT NAME - 3RD CHAR.  |  ACCT NAME - 4TH CHAR. | 33
   |--------------------------------------------------|
28 |  ACCT NAME - 5TH CHAR.  |  ACCT NAME - 6TH CHAR. | 34
   |--------------------------------------------------|
29 |  ACCT NAME - 7TH CHAR.  |  ACCT NAME - 8TH CHAR. | 35
   |--------------------------------------------------|
30 |                                                  | 36  FCBSTART
   |           START OF FILE BLOCK NUMBER             |
31 |                                                  | 37
   |--------------------------------------------------|
32 |                                                  | 40  FCBEND
   |    CURRENT NUMBER OF DATA BLOCKS IN THE FILE     |
33 |                                                  | 41
   |--------------------------------------------------|
34 |                                                  | 42  FCBNUM-
   | NUMBER OF OPEN AND CLOSE RECORDS (MESSAGE FILE)  |     OPENCLSREC
35 |                                                  | 43
   |--------------------------------------------------|
36 | LOGICAL DEVICE NUMBER  |                         | 44  FCBEXTMAP
   |------------------------                         |
37 |          FIRST EXTENT SECTOR NUMBER              | 45
   |--------------------------------------------------|
   |                        .                         |
   |                        .                         |
```

```
|                        .                         |
|--------------------------------------------------|
| LOGICAL DEVICE NUMBER  |                         |
|------------------------                          |
|            LAST EXTENT SECTOR NUMBER             |
----------------------------------------------------
```

Other identifiers used:

FCBSIZE	= FCB.(2:14)#,	size in words
FCBLKST	= FCB(4).(0:2)#,	previous lock state
FCBDTYPE	= FCB(4).(2:6)#,	device type
FCBCRUNCH	= FCB(4).(8:1)#,	pending crunch disposition
FCBSUBTYPE	= FCB(4).(12:4)#,	device subtype
FCBOCNTOUT	= FCB(5).(0:8)#,	no. accessors - output
FCBOCNT	= FCB(5).(8:8)#,	no. accessors
FCBCLASSFLG	= FCB(9).(0:1)#,	PV class flag
FCBMVTABX	= FCB(9).(4:4)#,	Mounted volume table index
FCBVMASK	= FCB(9).(8:8)#,	Volume Mask
FCBLBLEOF	= FCB(16).(0:8)#,	no. labels written
FCBLBL	= FCB(16).(8:8)#,	no. labels available
FCBBLKFACT	= FCB(18).(0:8)#,	blocking factor
FCBSECTPBLK	= FCB(18).(8:8)#,	sectors per block
FCBSECTOFF	= FCB(19).(0:8)#,	sector offset to data
FCBDISP	= FCB(19).(8:3)#,	pending disposition
FCBNUMEXTS	= FCB(19).(11:5)#,	no. extents less 1
FCBOCNTIN	= FCB(21).(8:8)#,	no. accessors - input
FCBLABEL	= FCBDBL(18)#,	label LDEV and sector
FCBLDEV	= FCB(36).(0:8)#,	label LDEV

Discussion:

FCBACBDST This is the DST of the ACB that was created at the same time as the FCB. This is used in conjunction with FCBNEWFCBDST when relocating the FCB.

FCBACBV This is the vector table entry of the ACB that was created at the same time as the FCB. This is used in conjunction with FCBNEWFCBV when relocating the FCB.

FCBAN This is the account name of the file. It is eight bytes in length with trailing blanks added.

FCBBLKFACT This is the blocking factor of the file. It is the number of logical records in a physical block. Legal values range from 1 to 255.

FCBDEVICE This specifies the device on which the file resides. If it is positive then it represents a logical device number; if negative it represents a (negative) device class index.

FCBDISP This is the pending FCLOSE disposition for the file. Legal values are:

0 - no change
1 - save permanent
2 - save temporary and rewind
3 - save temporary but do not rewind
4 - release
7 - invalid file (file label access error)

FCBCRUNCH This bit governs if space will be returned beyond the EOF upon the last FCLOSE of the file.

0 - no change
1 - return space beyond EOF

FCBDTYPE This is the device type number of the first extent of the file. See ACBDTYPE for a list of legal values.

FCBEND Block number of the file's EOF, relative to FCBSTART.

FCBEOF This is the end-of-file pointer for the file. It is a double integer representing the number of records in the file. It can also be viewed as the record number of the next record past EOF.

FCBEXCLSTAT This is the exclusive status of the file access. If -1 then the file is being accessed exclusively; otherwise it is the number of semi-exclusive accessors.

FCBEXTMAP This is the extent map of the file. The number of extents is specified by FCBNUMEXTS; a 0D extent descriptor indicates that the extent has not been allocated.

FCBEXTSIZE This is the extent size, in sectors, of the file. All extents in the file except possibly the last have this size. This is a logical value, and legal values range from 1 to 65535 sectors. This restricts the maximum file size to 2097120 sectors (268,431,360 words).

FCBFLIM This is the end-of-space pointer for the file. It is a double word integer representing the maximum number of records (fixed length record format) or blocks (undefined or variable length record format) in the file.

FCBFOPTIONS This is the FOPTIONS in effect for the file.

FCBGN This is the group name of the file. It is eight bytes long with trailing blanks added

FCBLABEL This is the logical device and sector number of the file label, which is the same as the first extent descriptor.

FCBLASTEXTSIZE This is the size, in sectors, of the last extent in the file. If the file has one extent then this is the same as FCBEXTSIZE; otherwise this value may be different from FCBEXTSIZE. This is the size of the last physical extent for the file; it is not the size of the last allocated extent.

FCBLBL This is the number of user labels allocated for the file. Since each label is a sector long, this is also the number of sectors allocated for user labels.

FCBLBLEOF This is the end-of-data pointer for the user labels. It is analogous to FCBEOF in that it represents the number of labels written. The initial value is 0.

FCBLDEV This is the logical device number of the first extent of the file.

FCBLKST This is the previous lock state of the file and is derived from the file label. Legal values are:

0 - no accessors
1 - read
2 - write
3 - read/write

FCBMVTABX If the file resides on a private volume, then this field represents the mounted volume table index of the volume set entry on which the file resides.

FCBNEWFCBDST This is the DST of the new FCB for the file. It is used in conjunction with FCBACBDST to move the FCB to a system (shared FCB) control block table when the second accessor is established. If this value is zero then there is no new FCB; if nonzero then a new FCB has been created.

FCBNEWFCBV This is the vector table entry of the new FCB for the file. It is used in conjunction with FCBACBV to move the FCB to a system (shared FCB) control block table when the second accessor is established. If this value is zero then there is no new FCB; if nonzero then a new FCB has been created.

FCBNUMEXTS This is the maximum number of extents, less one, allowed for the file. It is not the number of extents presently allocated, which is always determined by counting nonzero entries in the extent map.

FCBNUMOPENCLSREC Number of open and close records in the message file.

FCBOCNT This is the number of accessors for the file. Alternatively it can be viewed as the number of PACBs created for the file.

FCBOCNTIN This is the number of file accessors having input access.

FCBOCNTOUT This is the number of file accessors having output access.

FCBRIN This is the RIN number used to support dynamic locking (i.e. FLOCK and FUNLOCK) for the file. If there is no dynamic locking then this number is zero.

FCBSECTOFF This is the sector offset from the file label to the first block of the file. This is not necessarily equal to FCBLBL+1 since an integral number of blocks are allocated for the file and user labels.

FCBSECTPBLK This is the number of sectors in a block for the file.

FCBSIZE This is the size, in words, of the complete FCB. It includes the extent map.

FCBSTART Block number of the file's start, excluding the file label block.

FCBSUBTYPE This is the device subtype number of the first extent.

FCBUSERLBL This field describes the user labels for the file. It consists of FCBLBL and FCBLBLEOF, described separately.

FCBVMASK If the file resides on a private volume set, this bit mask signifies which volume of the set in which the file resides. Bit 15 is on it resides on the first volume, bit 14 if on the second, etc.

File Label (FLAB)

The file label has the following format:

```
  0   1   2   3           7   8         12  13  14  15
 -----------------------------------------------------
| FILE NAME - 1ST CHAR.  |  FILE NAME - 2ND CHAR.     |  0  FLLOCNAME
|-----------------------------------------------------|
| FILE NAME - 3RD CHAR.  |  FILE NAME - 4TH CHAR.     |  1
|-----------------------------------------------------|
| FILE NAME - 5TH CHAR.  |  FILE NAME - 6TH CHAR.     |  2
|-----------------------------------------------------|
| FILE NAME - 7TH CHAR.  |  FILE NAME - 8TH CHAR.     |  3
|-----------------------------------------------------|
| GROUP NAME - 1ST CHAR. | GROUP NAME - 2ND CHAR.     |  4  FLGRPNAME
|-----------------------------------------------------|
| GROUP NAME - 3RD CHAR. | GROUP NAME - 4TH CHAR.     |  5
|-----------------------------------------------------|
| GROUP NAME - 5TH CHAR. | GROUP NAME - 6TH CHAR.     |  6
|-----------------------------------------------------|
| GROUP NAME - 7TH CHAR. | GROUP NAME - 8TH CHAR.     |  7
|-----------------------------------------------------|
|  ACCT NAME - 1ST CHAR. |  ACCT NAME - 2ND CHAR.     | 10  FLACCTNAME
|-----------------------------------------------------|
|  ACCT NAME - 3RD CHAR. |  ACCT NAME - 4TH CHAR.     | 11
|-----------------------------------------------------|
|  ACCT NAME - 5TH CHAR. |  ACCT NAME - 6TH CHAR.     | 12
|-----------------------------------------------------|
|  ACCT NAME - 7TH CHAR. |  ACCT NAME - 8TH CHAR.     | 13
|-----------------------------------------------------|
| CREATOR NAME - 1ST CHAR.| CREATOR NAME - 2ND CHAR.  | 14  FLUSERID
|-----------------------------------------------------|
| CREATOR NAME - 3RD CHAR.| CREATOR NAME - 4TH CHAR.  | 15
|-----------------------------------------------------|
| CREATOR NAME - 5TH CHAR.| CREATOR NAME - 6TH CHAR.  | 16
|-----------------------------------------------------|
| CREATOR NAME - 7TH CHAR.| CREATOR NAME - 8TH CHAR.  | 17
|-----------------------------------------------------|
|  LOCKWORD - 1ST CHAR.  |  LOCKWORD - 2ND CHAR.      | 20  FLLOCKWORD
|-----------------------------------------------------|
|  LOCKWORD - 3RD CHAR.  |  LOCKWORD - 4TH CHAR.      | 21
|-----------------------------------------------------|
|  LOCKWORD - 5TH CHAR.  |  LOCKWORD - 6TH CHAR.      | 22
|-----------------------------------------------------|
|  LOCKWORD - 7TH CHAR.  |  LOCKWORD - 8TH CHAR.      | 23
|-----------------------------------------------------|
|                                                     | 24  FLSECMX
|                  SECURITY MATRIX                    |
|                                                     | 25
|-----------------------------------------------------|
| FILE LANGUAGE ATTRIBUTE |                | SR | S   | 26
|-----------------------------------------------------|
```

File Label (Cont.)

```
|                  CREATION DATE                       | 27  FLCREATE
|------------------------------------------------------|
|                 LAST ACCESS DATE                     | 30  FLLASTACC
|------------------------------------------------------|
|              LAST MODIFICATION DATE                  | 31  FLLASTMOD
|------------------------------------------------------|
|                    FILE CODE                         | 32  FLFILECODE
|------------------------------------------------------|
| C |              | MVTABX  |           VMASK         | 33 FLPVINFO
|------------------------------------------------------|
| S | R | L | X | SUBTYPE |    DISC TYPE     |  R/W    | 34  FLLOCK
|------------------------------------------------------|
| NO. USER LABELS WRITTEN | NO. USER LABELS AVAIL.     | 35  FLUSERLBL
|------------------------------------------------------|
|                                                      | 36  FLFLIM
|              FILE LIMIT IN BLOCKS                    |
|                                                      | 37
|------------------------------------------------------|
|                                                      | 40  FLFCBVECT
|                   FCB VECTOR                         |
|                                                      | 41
|------------------------------------------------------|
|                    CHECKSUM                          | 42  FLCHECKSUM
|------------------------------------------------------|
|                  COLD LOAD ID                        | 43  FLCLID
|------------------------------------------------------|
|                    FOPTIONS                          | 44  FLFOPTIONS
|------------------------------------------------------|
|               RECORD SIZE IN BYTES                   | 45  FLRECSIZE
|------------------------------------------------------|
|               BLOCK SIZE IN WORDS                    | 46  FLBLKSIZE
|------------------------------------------------------|
|     SECTOR OFFSET       |      |   NO. EXTENTS -1    | 47
|------------------------------------------------------|
|           LAST EXTENT SIZE IN SECTORS                | 50  FLLASTEXT-
|------------------------------------------------------|        SIZE
|             EXTENT SIZE IN SECTORS                   | 51  FLEXTSIZE
|------------------------------------------------------|
|                                                      | 52  FLEOF
|               END OF DATA POINTER                    |
|                                                      | 53
|------------------------------------------------------|
|  VOLUME TABLE INDEX    |                             | 54  FLEXTMAP
|-------------------------                             |
|            1ST EXTENT SECTOR NUMBER                  | 55
|------------------------------------------------------|
```

File Label (Cont.)

```
|                       .                        |
|                       .                        |
|                       .                        |
|------------------------------------------------|
| VOLUME TABLE INDEX    |                        |
|------------------------                        |
|           LAST EXTENT SECTOR NUMBER            |
|------------------------------------------------|
|                       .                        |
|                       .                        |
|                       .                        |
|------------------------------------------------|
|                                                |154  FLALLOCTIME
|          FILE ALLOCATION TIME                  |
|                                                |155
|------------------------------------------------|
|          FILE ALLOCATION DATE                  |156  FLALLOCDATE
|------------------------------------------------|
|                       .                        |
|------------------------------------------------|
|                                                | 160 FLSTART
|          START OF FILE BLOCK NUMBER            |
|                                                | 161
|------------------------------------------------|
|                                                | 162 FLEND
|     BLOCK NUMBER OF END OF FILE                |
|                                                | 163
|------------------------------------------------|
|                                                | 164 FLNUMOPENCLSREC
| NUMBER OF OPEN AND CLOSE RECORDS (MESSAGE FILE) |
|                                                | 165
|------------------------------------------------|
| DEVICE NAME - 1ST CHAR. | DEVICE NAME - 2ND CHAR. |174  FLDEVNAME
|------------------------------------------------|
| DEVICE NAME - 3RD CHAR. | DEVICE NAME - 4TH CHAR. |175
|------------------------------------------------|
| DEVICE NAME - 5TH CHAR. | DEVICE NAME - 6TH CHAR. |176
|------------------------------------------------|
| DEVICE NAME - 7TH CHAR. | DEVICE NAME - 8TH CHAR. |177
--------------------------------------------------
```

Other identifiers used:

FLSECURE	= FLAB(22).(15:1)#,	file secure bit
(FLSRRELEASE)	= FLAB(22).(14:1)#,	STORE/RESTORE released bit
FLCLASSFLG	= FLPVINFO.(0:1)#,	Class flag bit
FLMVTABX	= FLPVINFO.(4:4)#,	Mounted volume table index
FLVMASK	= FLPVINFO.(8:8)#,	Volume mask
(FLSTORE)	= FLAB(28).(0:1)#,	file being stored

FLRESTORE	= FLAB(28).(1:1)#,	file being restored
(FLLOAD)	= FLAB(28).(2:1)#,	file loaded
FLEXCL	= FLAB(28).(3:1)#,	exclusive access
FLSR	= FLAB(28).(0:2)#,	S & R bits
FLSRL	= FLAB(28).(0:3)#,	S, R, & L bits
(FLSRLX)	= FLAB(28).(0:4)#,	S, R, L, & X bits
FLSUBTYPE	= FLAB(28).(4:4)#,	device subtype
FLDTYPE	= FLAB(28).(8:6)#,	device type
FLSTATUS	= FLAB(28).(14:2)#,	write/read status
(FLLBLEOF)	= FLAB(29).(0:8)#,	no. labels written
(FLLBL)	= FLAB(29).(8:8)#,	no. labels available
FLSECTOFF	= FLAB(39).(0:8)#,	sector offset to data
FLNUMEXTS	= FLAB(39).(11:5)#,	no. extents less 1
FLLABEL	= FLABDBL(22)#,	label VTAB and sector
FLVTAB	= FLAB(44).(0:8)#,	label VTAB index

Discussion:

FLACCTNAME — This is the account name of the file. It is eight bytes in length with trailing blanks added.

FLALLOCDATE — Date that the file was allocated on this system.

FLALLOCTIME — Doubleword containing the time that the file was allocated on this system.

FLBLKSIZE — This is the block size, in sectors, of the file.

FLCHECKSUM — This is the exclusive-OR checksum of the file label (excluding words 34, 42, and 43 octal) and is used for error detection. Each time the file label is read from disc the check sum is calculated and compared against the value recorded in the file label. Similarly, each time the file label is written to the disc the check sum is calculated and inserted into the file label.

FLCLID — This is the cold load number in effect the last time that the file was accessed. This should always be the current cold load number. If it is not, it means that the system crashed while the file was open and that the data in the file label should be "reset" (principally the FCB vector FLFCBVECT).

FLCREATE — This is the creation date of the file. It is in the format defined by the intrinsic CALENDAR.

FLDEVNAME — This is the FOPEN device specification that was used when the file was created. This information is needed when new extents are allocated.

FLDTYPE — This is the device type number of the first extent of the file; see ACBDTYPE for a list of legal values. This value is determined by configuration.

FLEND — Number of current data blocks (that is, the end of file block number relative to the start of file).

FLEOF — This is the end-of-file pointer for the file. It is a double word integer representing the number of records in the file. It can also be viewed as the record number of the next record past EOF.

FLEXCL — This is the exclusive access flag for the file. If set it means that the file has been opened exclusively by a single accessor. If not set then the file is potentially accessible by others.

FLEXTMAP — This is the extent map of the file. The number of extents is specified by FLNUMEXTS; a 0D extent descriptor indicates that the extent has not been allocated.

FLEXTSIZE — This is the extent size, in sectors, of the file. All extents in the file, except the last, have this extent size. This is a logical value, and legal values range from 1 to 65535 sectors. This limits the maximum file size to 2097120 sectors.

FLFCBVECT — If nonzero, this is the vector of the FCB for the file. If zero, the file is not being accessed.

FLFILECODE — This is the file code of the file. Known values are:

1024	User Subprogram Library
1025	Basic Data
1026	Basic Program
1027	Basic Fast Program
1028	Relocatable library
1029	Program File
1031	Segmented Library
1035	View Form File
1036	View Fast Forms File
1037	View Reformat File
1040	Cross Loader ASCII File (SAVE)
1041	Cross Loader Relocated Binary File
1042	Cross Loader ASCII File (DISPLAY)
1050	Edit Quick File
1051	Edit KEEPQ File (COBOL)
1052	Edit TEXT File (COBOL)
1054	TDP Diary File
1055	TDP Proof Marked QMARKED
1056	TDP Proof Marked non-COBOL File
1057	TDP Proof Marked COBOL File
1058	TDP Workfile
1059	TDP Workfile (COBOL)
1060	RJE Punch File
1070	QUERY Procedure File
1080	KSAM Key File

1083	GRAPH Specification File
1084	User Logging Log File
1090	Self-describing File
1100	HPWORD Document
1101	HPWORD Hyphenation dictionary
1102	HPWORD Configuration File
1103	HP 2601 Environment File
1110	IDS/3000 Character Cell File
1111	IDS/3000 Form File
1112	IFS/3000 Environment File
1114	Graphics Image in RASTR Format
1130	OPT/3000 Log File
1131	TEPE/3000 Script File
1132	TEPE/3000 Log File
1133	APS/3000 Log File
1139	MPEDCP/DRP Log File
1140	HPToolset Root File
1141	HPToolset Data File
1145	Drawing File for HPDRAW
1146	Figure File for HPDRAW
1147	Reserved
1148	Reserved
1149	Reserved
1152	Compressed SLATE File
1153	Expanded SLATE Workfile
1156	Store File for RAPID/3000 Utility DICTDBU
1157	Code File for Transact/3000 Compiler
1158	Code File for Report/3000 Compiler
1159	Code File for Inform/3000 Compiler
1166	HPDESK Distribution list
1167	HPDESK Text
1177	Term Type File
1178	Term Vertical Format Control File
1192	Network Configuration File
1193	Network Trace File
1194	Network Log File
1211	Reserved
1212	Reserved
1226	VC File
1227	DIF File
1228	Language Definition File
1229	Character Set Definition File
1230	Formatted Application Message Catalog
1235	Reserved
1236	Reserved
1258	Pathflow STATIC File
1259	Pathflow DYNAMIC File
8000 to 8099	Reserved for APL

FLFLIM — This is the end-of-space pointer for the file. It is a double integer representing the maximum number of records (fixed length record format) or blocks (undefined or variable length record format) in the file.

FLFOPTIONS This is the FOPTIONS of the file.

FLGRPNAME This is the group name of the file. It is eight bytes long with trailing blanks added.

FLLABEL This is the volume table index and sector number of the file label, which is the same as the first extent descriptor. FLLASTACC This is the last access date of the file. It is in the format defined by the intrinsic CALENDAR.

FLLASTMOD This is the last modification date of the file. It is in the format defined by the intrinsic CALENDAR.

FLLASTEXTSIZE This is the size, in sectors, of the last extent in the file. If the file has one extent, then this is the same as FLEXTSIZE; if the file has more than one extent, then this value may be different from FLEXTSIZE. This is the size of the last physical extent for the file; it is not the size of the last allocated extent.

FLLBL This is the number of user labels allocated for the file. Since each label is a sector long, this is also the number of sectors allocated for user labels.

FLLBLEOF This is the end-of-data pointer for the user labels. It is analogous to FLEOF in that it represents the number of labels written.

FLLOAD This is the LOADED flag for the file. If set, it means that the file is a loaded program or SL file and cannot be modified except by a privileged accessor. This flag is set and cleared by the loader, not the file system.

FLLOCK This identifies the word containing the lock bits, which are described separately.

FLLOCKWORD This is the lock word of the file. It is eight bytes long with trailing blanks added. If it is all blanks, then the file does not have a lockword.

FLLOCNAME This is the local name of the file. It is eight bytes long with trailing blanks added.

FLNUMEXTS This is the number of extents, less one, allowed for the file. It is not the number of extents allocated. Legal values range from 0 to 31, i. e., 1 to 32 extents.

FLNUMOPENCLSREC Number of open and close records in the message file.

FLPVINFO File label private volume information. This is in the same format as the FCBPVINFO.

FLRECSIZE This is the record size of the file in negative bytes.

FLRESTORE This is the RESTORE flag for the file. If set, it means that the file is being RESTOREd and cannot be accessed. RESTORE also sets the STORE bit for the file (FLSTORE); see FLSR for a full description of the use of these bits. This flag is set and cleared by STORE/RESTORE, not the file system.

FLSECMX This is the security matrix of the file. The bits are organized into five groups of six bits each. (Bits 0:2 are not used.) The groups correspond to the access types: READ, APPEND, WRITE, LOCK, and EXECUTE. Within each group, each bit specifies who may have the access: ANY, ACCOUNT MGR, ACCOUNT LIB- RARIAN, GROUP, GROUP LIBRARIAN, CREATOR.

FLSECTOFF This is the sector offset from the file label to the first block of the file. This is not necessarily equal to FLLBL+1 since an integral number of blocks are allocated for the file and user labels.

FLSECURE This is the file security enforcement flag for the file. If not set, then the file has been RELEASEd and the security matrix FLSECMX should be ignored. If set, then secure as specified by the security matrix.

FLSR This is the STORE and RESTORE flags for the file, which are described separately. STORE and RESTORE decode the two-bit field to indicate their operation. Legal values are:

0 - file not in use by either STORE or RESTORE
1 - illegal value
2 - file being STOREd
3 - file being RESTOREd

The file system interprets the leftmost bit as indicating that the file is being accessed by either STORE or RESTORE. The rightmost bit is interpreted as indicating what access should be permitted: 0 (file being STOREd) allows read access; 1 (file being RESTOREd) allows no access. This field is set and reset by STORE/RESTORE, not the file system.

FLSRL This is the STORE, RESTORE and LOADED flags for the file, which are described separately.

FLSRLX This is the STORE, RESTORE, LOADED and exclusive flags for the file, which are described separately.

FLSRRELEASE This flag is used by STORE/RESTORE. If a file is STOREd with the ";RELEASE" keyword, STORE will set this flag

in the tape copy of the file label. RESTORE will allow any user to access such files, regardless of the file's normal security. If this bit is off in the tape copy of the file label, RESTORE applies normal security checks (as defined by the information in FLSECMX and FLSECURE). This bit is zero for files on disc.

FLSTART Block number of the file's start, excluding the file label block.

FLSTATUS This is the read/write status of the file. Legal values are:

0 - no accessors
1 - read
2 - write
3 - read/write

FLSTORE This is the STORE/RESTORE flag for the file. If set it means that the file is being either STOREd or RESTOREd. The RESTORE bit (FLRESTORE) must be interrogated to determine which operation is taking place; see FLSR for a full description of the use of these bits. This flag is set and cleared by STORE/RESTORE, not the file system.

FLSUBTYPE This is the device subtype number of the first extent of the file. This value is determined by configuration.

FLUSERID This is the creating user name of the file. It is eight bytes long with trailing blanks added.

FLUSERLBL This field describes the user labels of the file. It consists of FLLBL and FLLBLEOF, which are described separately.

FLVTAB This is the volume table index of the first extent of the file.

File Multi-Access Vector Table (FMAVT) DST(%54)

The FMAVT is used to locate shared PACB's for files opened multi-access. When an old disc file has been opened multi-access, the FMAVT is searched to determine if the file has previously been opened. The JITDST and the DADDR found in the FMAVT are compared to the JITDST of the job and the DADDR of the device or disc file being opened multi-access. If an entry exists for the file, then the PACB can be easily located for that file. If this is the first process opening the file, then an entry is created and inserted into the FMVAT for the file.

Spoolfiles are opened multi-access, therefore, they will have entries in the FMAVT. $STDIN and $STDLIST also have entries in the FMAVT since they too are opened multi-access.

Zero Entry Format

CURRENT TABLE SIZE	0 FM'CURR'SIZE
ENTRY SIZE = 6	1 FM'ENTRY'SIZE
MAXIMUM TABLE SIZE	2 FM'MAX'SIZE
0	3
0	4
0	5

Descriptions:

FM'CURR'SIZE The current size of the FMAVT in words. This value increases in increments of %200 words until FM'MAX'SIZE is reached.

FM'MAX'SIZE The maximum allowable size in words that the FM'CURR'SIZE can get. The current value of this is %4000. FM'MAX'SIZE can be changed only by changing the code in Initial. The open of the multi-access file is failed if this maximum is reached.

FM'ENTRY'SIZE Size in words of an FMAVT entry, 6 words at present.

Typical Entry Format

```
  0   1   2   3     6   7   8        12  13  14  15
 ---------------------------------------------------
| 1 | G | D |     |                UNUSED          | 0
|--------------------------------------------------|
|                    JIT DST                       | 1 FM'JITDST
|--------------------------------------------------|
|     LOGICAL DEVICE       |                       | 2 FM'DADDR
|--------------------------                        |
|                  DISK ADDRESS                    | 3
|--------------------------------------------------|
|                                                  | 4 FM'PACBV
|                  PACB VECTOR                     |
|                                                  | 5
 ---------------------------------------------------
```

```
FM'DEVICE    = FMAVT(0).(2:1)#,    Device bit
FM'GLOBAL    = FMAVT(0).(1:1)#,    Global multi-access bit
FM'LDEV      = FM'DADDR(0).(0:8)#, Logical device number of file
```

Descriptions:

FM'DADDR The disc address of the file label for disc files. For device files, the disc address is zero.

FM'DEVICE This bit is 1 for device files and 0 for disc files.

FM'LDEV Logical device number of device files or the LDEV of the disc containing the file label for disc files.

FM'JITDST The DST number of the JIT for the job that has the file open. If this field is nonzero, then only processes in the family tree of this particular job can open the file. This field is zero if the file was open global multi-access.

FM'GLOBAL This bit is 1 if the file was opened global multi-access, this allows multi-access to the file between jobs.

FM'PACBV The PACB vector for this multi-access file. Used to easily find the Physical Access Control Block for files opened multi-access.

System Global Area (SYSGLOB)

The file system uses several words in the system global area for its own use.

```
SHFCBDST     = SYSDB+%76,      shared CBT DST no.
MONITOR      = SYSDB+%77,      monitoring flag word
MAXSSECT     = SYSDB+%100,     max # spoolfile sectors
NUMSSECT     = SYSDB+%102,     current # spoolfile sectors
EXTSSECT     = SYSDB+%104,     # sectors/spoolfile extent
SPOOLINDEX   = SYSDB+%132,     class spool index
CSIOWAIT     = SYSDB+%135,     CSIOWAIT PLABEL
CCLOSEPLABL  = SYSDB+%140,     CS CCLOSE PLABEL - FPROCTERM
DSCHKPLABL   = SYSDB+%335,     DSCHECK PLABEL
DSOPENPLABL  = SYSDB+%336,     DSOPEN PLABEL
DSCLOSEPLABL = SYSDB+%337,     DSCLOSE PLABEL
SDSLDEVLABEL = SYSDB+%323,     PLABEL for SDSLDEV
MANWCPLABL   = SYSDB+%340;     MANAGEWRITECONV PLABEL
GLOBALAFTDST = SYSGLBEXT+%121 Global AFT DST number
```

SIRs, Locks, and Deadlocks

The file system uses two SIRs: the File SIR, which is intended to protect file label integrity, and the FMAVT SIR, which is to guarantee the integrity of the FMAVT. Since the file system locks these resources and also locks control blocks, deadlocks can occur if locking is done in the wrong order. Not only must the file system handle locking correctly, but the entire ensemble of the file system, its callers, and its callees must do so also. These include KSAM, which has a SIR of its own, SYSDUMP, and STORE, which lock the File SIR because they tweak bits in file labels. The presently accepted order is:

Get FMAVT SIR Lock ACB Get File SIR Lock FCB

It may not be necessary to do all of these things in any particular procedure. In modifying a procedure, you should be sure that any of these locks which you change are consistent not only within your own code, but also with its callers and callees.

Shared CBT DST

In sysglobal %76 (ABSOLUTE %1076) there exists the shared Control Block Table DST number. This DST holds a list of shared CBT's. Shared CBT's are used to keep any and all file system control blocks that have the potential to be shared between processes. Any disc file opened shared will have its FCB kept in one of these CBT's. Also, all terminal PACB's will be stored in a system shared CBT so that an extra data segment is not wasted. This is possible because all terminal access is performed NOBUF, which means that the PACB will be a minimal PACB and can be placed in these CBTs. Lastly, any file opened with global file access will have all its control blocks placed into these system CBT's.

The format of the system shared CBT DST is similar to a Control Block Table. It has the same words of overhead and the data (the list of DST's) starts in the next word after the overhead. The system CBT's are created one at a time as needed. Usually, there are only a few DST's in the list.

```
|--------------------------------------------|
|         TABLE SIZE IN WORDS (%200)         |  0
|--------------------------------------------|
|         DST NUMBER OF THIS TABLE           |  1
|--------------------------------------------|
|                     0                      |  2
|--------------------------------------------|
|                     0                      |  3
|--------------------------------------------|
|                     0                      |  4
|--------------------------------------------|
|                     0                      |  5
|--------------------------------------------|
|                     0                      |  6
|--------------------------------------------|
|                     0                      |  7
|--------------------------------------------|
|        1ST. SHARED CBT DST NUMBER          |  10
|--------------------------------------------|
|        2ND. SHARED CBT DST NUMBER          |  11
|--------------------------------------------|
|                     .                      |
|                     .                      |
|                     .                      |
|--------------------------------------------|
|        118TH. SHARED CBT DST NUMBER        |  177
|--------------------------------------------|
```

CHAPTER 7 PROCESS TABLES

The operating system maintains state, control, and accounting information on each process. The data structures for this purpose are the process control block table (PCB; core resident, 1 entry per process) and the process control block extension (PCBX; contained in the process' stack below DL). Process related information which must be accessible when the process' stack is not present in main memory is maintained in the process' PCB entry. All other process related information is maintained in the process' PCBX.

A process is identified in the system by its PCB entry number, referred to as its PIN (process identification number), or by its PCBPT=(PIN)*(PCB entry size).

The structure of the PCB table, PCB entry format, PCBX structure, and PCBX format are specified in this chapter.

Process Control Block Table Structure and Format

Fixed Cells Related to PCB

4 PCB relative index of current process' PCB entry
%1003 Absolute address of the PCB table base
The bank & address are represented as per the MPEV ERS.
%1271 PCB relative address of head of dispatching queue's PCB entry
%1272 PCB relative address of tail of dispatching queue's PCB entry

PCB Entry 0 Format

0	# OF CONFIGURED ENTRIES
1	ENTRY LENGTH (%25)
2	# OF UNASSIGNED ENTRIES
3	TABLE RELATIVE INDEX TO FIRST UNASSIGNED ENTRY
4	TABLE RELATIVE INDEX OF LAST FREE ENTRY
5	HIGH WATER MARK
6	NUMBER OF PRIMARY CONFIGURED ENTRIES (0)
7	HEAD OF IMPEDED QUEUE PCB RELATIVE INDEX
8	TAIL OF IMPEDED QUEUE PCB RELATIVE INDEX
9	NUMBER OF CURRENTLY IMPEDED PROCESSES
10	NUMBER OF MAXIMUM IMPEDED PROCESSES (CURRENT)
11	CUMULATIVE NUMBER OF IMPEDED PROCESSES(CURRENT)
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0

Unassigned PCB Entry Format

0	0
1	TABLE RELATIVE INDEX TO NEXT UNASSIGNED ENTRY
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	%177777

Assigned PCB Entry Format

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
      |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
PCB00 |S |B |C |H |P |H |I |P |D |L |S |T |U |H |S |R |
      |A |F |R |S |I |S |P |C |S |W |W |R |S |I |T |I |
      |R |  |I |I |O |P |E |  |O |  |  |W |E |P |O |T | RESABORTINFO
      |  |  |T |R |V |R |X |  |F |  |  |  |D |R |V |B |
      |  |  |  |  |R |I |P |  |T |  |  |  |Q |I |A |K |
      |-----------------------------------------------|
PCB01 |  SLL RELATIVE ADDRESS OF PROCESS' SEGMENT      | SLLPTR
      |               LOCALITY LIST                   |
      |-----------------------------------------------|
      | A| /|                                         |
PCB02 | D| /|         XDS      DST#                   | DBXDSINFO
      | B| /|                                         |
      |-----------------------------------------------|
      | A| S|                                         |
PCB03 | O| C|         STK      DST#                   | STKINFO
      |-----------------------------------------------|
      |  |  |  |  | B|  | U| J| T| M| S|  | I| S| T| M|
PCB04 | M| R| R| M| I| I| C| N| I| S| O|FA| M| I| I| E| WAKEMASK
      |  | G| L| A| O| O| P| K| M| G| N|  | P| R| M| M|
      |-----------------------------------------------|
PCB05 |              FATHER'S  PCB INDEX              | FATHERINFO
      |-----------------------------------------------|
PCB06 |               SON'S PCB INDEX                 | SONINFO
      |-----------------------------------------------|
PCB07 |              BROTHER'S PCB INDEX              | BROTHERINFO
      |-----------------------------------------------|
      |        |W |     | D|  |                      |
PCB08 |        |S |     | E|F |                      | PIINFONIMPPIN
      |  PSIM  |O | OA  | A|A |//////////////////////|
      |        |F |     | D|C |                      |
      |        |T |     |  |  |                      |
      |-----------------------------------------------|
PCB09 |L | BMS | PPC |S |  PTYPE |S |HK|SK|ST|HB|CY|BK| PROCSTATE
      |I |     |     |O |        |I |  |  |  |  |  |  |
      |V |     |     |V |        |  |  |  |  |  |  |  |
      |-----------------------------------------------|
PCB10 |                EVENT FLAGS                 |WS| EVENTFLAGS
      |-----------------------------------------------|
PCB11 |       SEGIDENTIFIER OF LAST REFERENCED        | LASTREFSWAPSEG
      |-                                             -|
PCB12 |           SWAPPABLE CODE SEGMENT              |
      |-----------------------------------------------|
PCB13 |D |L |C |D |E |I |C |A |                      | QUEUEINGINFO
      |I |Q |  |  |  |N |O |S |                      |
      |S |  |  |  |  |T |R |O |      PRIORITY        |
      |P |  |  |  |  |E |E |F |                      |
      |Q |  |  |  |  |R |R |T |                      |
      |-----------------------------------------------|
```

Assigned PCB Entry Format (Cont.)

```
PCB14|                BLKINX                      | PBX
     |--------------------------------------------|
PCB15|          CST MAPPING DST #                 | MAPDST
     |--------------------------------------------|
PCB16|             PIMP PCB INDEX                 | PIMPPIN
     |--------------------------------------------|
PCB17|             NIMP PCB INDEX                 | NIMPPIN
     |--------------------------------------------|
PCB18|               BPTLINK                      | BPTLINK
     |--------------------------------------------|
PCB19|  PCB INDEX OF NEXT PCB ENTRY IN QUEUE       | NQPTR
     |--------------------------------------------|
PCB20| PCB INDEX OF PREVIOUS PCB ENTRY IN QUEUE   | PQPTR
     |--------------------------------------------|
```

```
PCB00  .(0:1)   SAR ==> scheduling attention required
       .(1:1)   Bounds Flag -- Privilege mode bounds check
       .(2:1)   CRIT ==> process is critical
       .(3:1)   HSIR ==> process has a sir
       .(4:1)   PIOVR ==> pending PI, process critical
       .(5:1)   HSPRI ==> hold sir priority
       .(6:1)   IPEXP ==> incore protect expired
       .(7:1)   PC ==> pre-empt capability
       .(8:1)   DSOFT ==> Delayed soft int processing.  A pending
                          soft int cannot be processed because of sir
                          or critical state.  PSEUDOINT will be invoked
                          when these condition(s) go away.
       .(9:1)   LW ==> long wait
       .(10:1)  SW ==> short wait
       .(11:1)  TRW ==> terminal read wait
       .(12:1)  USEDQ ==> used a quantum since transaction began
       .(13:1)  HIPRI ==> hold impeded priority
       .(14:1)  STOVA ==> processing abort due to stack overflow.
       .(15:1)  RITBK==> Request Information Table Break

PCB01  .(0:16)  SLLPTR, SLL relative index to process' segment
                locality list

PCB02  .(0:1)   ADB, set if DB pointing to an absolute address
       .(2:14)  XDS, DST entry number of extra data segments to which
                DB is set; zero if none.

PCB03  .(0:1)    STOVALL FLAG ==> stack overflow is already allocated
       .(1:2)    SC, set if executing system code
       .(2:14)   DST entry number of process' stack

PCB04  .(0:1)    M, mourning wait.
       .(1:1)    RG, global RIN wait.
       .(2:1)    RL, local RIN wait.
```

```
       .(3:1)   MA, mail wait.
       .(4:1)   BIO, blocked I/O wait.
       .(5:1)   IO, I/O wait.
       .(6:1)   UCP, UCOP wait and RIT wait.
       .(7:1)   JNK, junk wait.
       .(8:1)   TIM, timer wait.
       .(9:1)   MSG, file system basic IPC message wait.
       .(10:1)  SON, son wait.
       .(11:1)  FA, father wait.
       .(12:1)  IMP, process waiting to be unimpeded.
       .(13:1)  SIR, process waiting for a sir.
       .(14:1)  TOT, process waiting for a time out.
       .(15:1)  MEM, process waiting for memory.

PCB05  .(0:16)  FPIN, father's PCB relative index

PCB06  .(0:16)  SPIN, son's PCB relative index

PCB07  .(0:16)  BPIN, brother's PCB relative index

PCB08  .(0:3)   PSIM, pseudo - interrupt mode
                  1: hard kill
                  2: soft kill
                  3: stop
                  4: hibernate
                  5: escape
                  6: break
                  7: normal
       .(3:1)   ASOFT, ON for soft interrupt to wake process
                even though it is waiting on another event.
       .(4:2)   BR-
                 0: other source
                 1: Father
                 2: son
                 3: reply done on RIT wait
       .(6:1)   DEAD, set during expiration.
       .(7:1)   FAC, if set, the father is to be activated on process
                termination.

PCB09  .(0:1)   ALIVE, set if process is alive.
       .(1:2)   BMS, block mail, valid if MA set
                 0: sent to father
                 1: received from father
                 2: send to son
                 3: received son
       .(3:2)   PPC, process to process communication, set with
                respect to son.
                 0: null
                 1: son to father
                 2: Father to son
                 3: blocked
       .(5:1)   STOV, stack overflow bit
       .(6:3)   PTYPE, process type
                 0: user
```

```
                 1:  user, son of main
                 2:  user, main
                 3:  user, main, task
                 4:  system
                 5:
                 6:  system, UCOP
                 7:
       .(9:1)   SI, set when the Dispatcher (and PSEUDOINT)
                should be aware of a pending soft interrupt.
       .(10:1)  HK, hard kill pseudo interrupt
       .(11:1)  SK, soft kill pseudo interrupt
       .(12:1)  ST, stop pseudo interrupt
       .(13:1)  HB, hibernate pseudo interrupt
       .(14:1)  CY, control-y pseudo interrupt
       .(15:1)  BK, break pseudo interrupt

PCB10  .(0:15)  EVENTFLAGS, one for each wait class in PCB04
       .(15:1)  WS, wake up waiting switch set if an awake is
                missing.

PCB11  .(0:32)  LASTREFSWAPSEG, segment identifier of last
                referenced swappable code segment.

PCB13           (QUEUING INFO)
       .(0:1)   DISPQ ==> on dispatching queue
       .(1:1)   L scheduling class
       .(2:1)   C scheduling class
       .(3:1)   D scheduling class
       .(4:1)   E scheduling class
       .(5:1)   INTER ==> process is interactive
       .(6:1)   CORER ==> process is core resident
       .(7:1)   ASOFT, Allow soft interrupt.  A value of 1
                implies that user soft interrupts will be
                processed.  A zero value inhibits user soft
                ints (they are queued).  This bit is managed
                by FINTSTATE and FINTEXIT intrinsics.
       .(8:8)   Process' scheduling priority

PCB14  .(0:16)  PBX, CSTX block map index of process' program.

PCB15  .(0:16)  MAPDST, DST entry number of the CST mapping
                table.

PCB16  .(0:16)   PIMPPIN, PCB relative index of previous impeded PIN.

PCB17  .(0:16)   NIMPPIN, PCB relative index of next impeded PIN.

PCB18  .(0:16)   BPTLINK, breakpoint link for process

PCB19  .(0:16)   NQPTR, PCB relative index of next proc in disp queue

PCB20  .(0:16)   PQPTR,PCB relative index of prev proc in disp queue
                 queue
```

PCBX Structure and Format

PCBX General Structure

```
       ----------------------------  ------------------------
a--->| DL-a = SEG. REL DL VALUE    |                      ^
     |-----------------------------|        |             |
     | DB-a = SEG> REL DB VALUE    |        |             |
     |-----------------------------|        |             |
     |                             |     PXGLOB           |
     |                             |        |             |
     |                             |        |             |
     |-----------------------------|  -------------       |
b--->|  c-b = PXFIXED LENGTH       |                      |
     |-----------------------------|        |             |
     |                             |        |             |
     |                             |        |             |
     |-----------------------------|        |             |
     | PXFIXED EXPANSION AREA      |     PXFIXED          |
     |        BIT MAP              |        |             |
     |       (4 words)             |        |             |
     |-----------------------------|        |             |
     |                             |        |             |

          PXFIXED EXPANSION                 |             |
     |                             |                      |
     |                             |                    PCBX
     |-----------------------------|  -------------     SIZE
c--->|   d-c = PXFILE LENGTH       |                      |
     |-----------------------------|        |             |
     |                             |        |             |
     |-----------------------------|        |             |
     |                             |     PXFILE           |
                   |                        |             |
  PXFILE EXPANSION/CONTRACTION              |             |
     |             |               |        |             |
     |             |               |        |             |
     |-----------------------------|  -------------       |
d--->| COUNT OF SECTORS ALLOCATED  |                      |
     |  FOR PXFIXED EXPANSION      |                      |
     |-----------------------------|                      |
     |           DL-c              |                      |
     |-----------------------------|                      |
     |           DL-b              |                      |
     |-----------------------------|                      |
     |           DL-a              |                     \ /
     |-----------------------------|  ------------------------
DL-->|                             |


     |                             |
     |-----------------------------|
DB-->|                             |
```

PXGLOB Format

The PXGLOB portion of the pcbx is for job information, and contains the same job related information for all processes belonging to the same job.

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0|           DL-a=SEG. REL DL VALUE              |0
   |-----------------------------------------------|
  1|           DB-a=SEG. REL DB VALUE              |1
   |-----------------------------------------------|
  2|               USER ATTRIBUTES                 |2
   |-----------------------------------------------|
  3|                JMAT INDEX                     |3
   |-----------------------------------------------|
  4|               JPCNT INDEX                     |4
   |-----------------------------------------------|
  5|                JCUT INDEX                     |5
   |-----------------------------------------------|
  6|SB| R|  TY | D| I|//|//|//|//| STACK DUMP FLAGS|6
   |-----------------------------------------------|
  7|////////////////////////|     NATIVE LANGUAGE  |7
   |-----------------------------------------------|
 10|          ACTUAL JOB INPUT LDEV                |8
   |-----------------------------------------------|
 11|          ACTUAL JOB OUTPUT LDEV               |9
   |-----------------------------------------------|
 12|                JDT DST INDEX                  |10
   |-----------------------------------------------|
 13|                JIT DST INDEX                  |11
   |-----------------------------------------------|
```

```
 R = restart bit                      Stack Dump Flags
 I = job in/list interactive          Bit 10 = Armed
 D = job in/list duplicative          Bit 11 = Suppress traceback
TY = job type                         Bit 12 = Suppress ASCII
 0 = undefined                        Bit 13 = Q-63 to S
 1 = session                          Bit 14 = QINIT to S
 2 = job                              Bit 15 = DL to QINIT
 3 = task
 * = reserved:
 SB= stun bit ;used for stack underflow simulation for ICF44 or ICF55.
```

PXFIXED Assignments

The PXFIXED portion of the pcbx contains specific information and control information.

```
   |-----------------------------------------------|
  0|               c-b PXFIXED SIZE                |0
   |-----------------------------------------------|
  1|               RELATIVE S(S-DB)                |1
   |-----------------------------------------------|
  2|               RELATIVE Z(Z-DB)                |2
   |-----------------------------------------------|
  3|                INITIAL Q(Q-DB)                |3
   |-----------------------------------------------|  LM MDST existed
  4|         INITIAL RELATIVE DL (DB-DL)           |4 LP LOADPROCed
   |-----------------------------------------------|  Trap Modes
  5|   GENERAL RESOURCE CAPABILITY(FROM PROG-FILE) |5 .AT(0:1)-Arith.
   |-----------------------------------------------|  .LT(1:1)-Library
  6|AT|LT|ST|CY|CT|//|//|//|//|U |L |C |G |A |LM|LP|6 .ST(2:1)-System
   |-----------------------------------------------|  .CY(3:1)-Ctl-Y
  7|LINK TO XDS ENTRIES IN EXP. area   |   XDS CNT |7 .CT(4:1)-Code
   |-----------------------------------------------|  U User UDC exist
 10| P| S|   EXTRA DATA SEGMENT DST INDEX          |8 L Logging
   |-----------------------------------------------|  C Share Clock
 11| P| S|   EXTRA DATA SEGMENT DST INDEX          |9 G Global RIN acquired
   |-----------------------------------------------|  A Acct UDC exist
 12| P| S|   EXTRA DATA SEGMENT DST INDEX          |10  / 0:1 RESERVED FOR
   |-----------------------------------------------|    |      CST EXPANSION
 13| P| S|   EXTRA DATA SEGMENT DST INDEX          |11  | 1:1 = 1 IF ABORT
   |-----------------------------------------------|    |        IN PROGRESS
 14| X| A|   ABORT Y       |RW|  INITIAL CST INDEX |12  < 7:1 = 0 IF HAVE R/W
   |-----------------------------------------------|    |        ACCESS TO
 15|       MAXIMUM STACK SIZE(MAXDATA LIMIT)       |13  |        PROG FILE
   |-----------------------------------------------|    |      = 1 OTHERWISE
 16|             ARITHMETIC TRAP MASK              |14  | 8:8 = CST # OF SEG
   |-----------------------------------------------|    |   INITIALLY EXECUTED
 17|            ARITHMETIC TRAP PLABEL             |15  \       AT PROCCREATION
   |-----------------------------------------------|
 20|              LIBRARY TRAP PLABEL              |16
   |-----------------------------------------------|
 21|              SYSTEM TRAP PLABEL               |17
   |-----------------------------------------------|
 22|               CONTROL Y PLABEL                |18
   |-----------------------------------------------|
 23|               CODE TRAP PLABEL                |19
   |-----------------------------------------------|
 24|      DATA COM TERMINATION TRAP PLABEL         |20
   |-----------------------------------------------|
 25|              IMAGE TRAP PLABEL                |21
   |-----------------------------------------------|
 26|        RESERVED                               |22
   |-----------------------------------------------|
 27|CUR.MAX STACK SIZE(largest value ever for Z-DL)|23
   |-----------------------------------------------|
```

PXFIXED Assignments (Cont.)

```
   |-----------------------------------------------|
 30|               PROCESS CPU TIME                |24
   |                                               |
 31|                    (MSEC)                     |25
   |-----------------------------------------------|
 32|     MAXIMUM DATA SEG SIZE USED(IN SECTORS)    |26
   |-----------------------------------------------|
 33|    TOTAL VIRTUAL STORAGE USED(IN SECTORS)     |27
   |-----------------------------------------------|
 34|       CURRENT EXTRA DATA SEGMENT SPACE        |28
   |-----------------------------------------------|
 35|       MAXIMUM EXTRA DATA SEGMENT SPACE        | 29
   |-----------------------------------------------|
 36| PRIV MODE BOUNDS FLAGS|       STOV COUNT      | 30
   |-----------------------------------------------|
 37|  PROCESS EXECUTION TIME REMAINDER (IN MSEC)   | 31
   |-----------------------------------------------|
 40|        SET TO-1 WHEN IN BREAK MODE*           | 32
   |-----------------------------------------------|
 41|      CONTINUE FLAG (:CONTINUE COMMAND)**      | 33
   |-----------------------------------------------|
 42|ACTUAL SIZE OF VIRTUAL SPACE ALLOCATED TO STACK| 34
   |-----------------------------------------------|
 43|                 ERROR LEVEL                   | 35
   |-----------------------------------------------|
 44|              INTRINSIC ERRORS                 | 36
   |-----------------------------------------------|
 45|              INTRINSIC ERRORS                 | 37
   |-----------------------------------------------|
 46|              INTRINSIC ERRORS                 | 38
   |-----------------------------------------------|
 47|              INTRINSIC ERRORS                 | 39
   |-----------------------------------------------|
 50|              INTRINSIC ERRORS                 | 40
   |-----------------------------------------------|
 51|              INTRINSIC ERRORS                 | 41
   |-----------------------------------------------|
 52|TSLR, virtual time since last rescheduled      |42
   |-----------------------------------------------|
 53|TSTB, virtual time since transaction began     |43
   |-----------------------------------------------|
 54|TSSWAPIN, virtual time since swapin            |44
   |-----------------------------------------------|
 55|TSLA, virtual time since last absence          |45
   |-----------------------------------------------|
 56|TSLD, virtual time since last deallocation     |46
   |-----------------------------------------------|
 57|QCNT, quantums used since transaction began    |47
   |-----------------------------------------------|
```

PXFIXED Assignments (Cont.)

```
   60|/|D|/|O| RESERVED FOR FUTURE SOFT INT USE      |48
     |/|C|/|S|                                       |
     |/|Y|/|I|                                       |
     |-----------------------------------------------|
   61| TRLX INDEX FOR KERNEL TIMEOUT PROCEDURE       | 49
     |-----------------------------------------------|   JOB TYPE:
   62|TY |    JOB/SESSION NUMBER                     | 50   1=SESSION
     |                                               |      2=JOB
   63|<---(reserved )------------------------------->| 51
     |-----------------------------------------------|
   64|           RESERVED FOR FUTURE USE             | 52
     |-----------------------------------------------|
   65|           RESERVED FOR FUTURE USE             | 53
     |-----------------------------------------------|
   66|           RESERVED FOR FUTURE USE             | 54
     |-----------------------------------------------|
   67|           RESERVED FOR FUTURE USE             | 55
     |-----------------------------------------------|
   70|  |CY|  |SI|                                   | 56
     |-----------------------------------------------|
   71|               TIMEOUT TRLX                    | 57
     |-----------------------------------------------|
   72|///////////////////////////////////////////////| 58
     |-----------------------------------------------|
   73|///////////////////////////////////////////////| 59
     |-----------------------------------------------|
   74|     PCLASSMASK                                | 60
     |-----------------------------------------------|
   75|     PROCQUESTOPWORD                           | 61
     |-----------------------------------------------|
   76|                                               | 62
     |     PROCSTOPTIME                              |
   77|                                               | 63
     |-----------------------------------------------|

     |                  UNUSED                       |
     |-----------------------------------------------|
  114|                                               |
     |         PXFIXED EXPANSION BITMAP              |
  117|                                               |
     |-----------------------------------------------|
```

```
NOTES:  P = 1 if opened by priv user
        S = 1 if data segment is sharable

       PCLASSMASK    = BIT MASK OF CLASSES THIS PROCESS HAS ENABLED
       PROCQUESTOPWORD.(0:4) = PROCESS PRIORITY: 7 => L QUEUE
                                                 6 => C QUEUE
                                                 2 => D QUEUE
                                                 1 => E QUEUE
```

```
                .(4:12)= REASON STOPPED: 1 => STOP SEG FAULT
                                         2 => STOP DISC WAIT
                                         3 => BLOCKED I/O, NON TERMINAL
                                         4 => TERMINAL READ
                                         5 => STOP IMPEDE
                                         6 => STOP ACTIVE
       PROCSTOPTIME = DBL WORD TIMESTAMP OF WHEN PROCESS STOPPED FOR
                      REASON GIVEN IN PROCQUESTOPWORD
```

```
    DCY           A DELAYED CONTROL Y IS PENDING (THIS BIT
                  IS CHECKED BY ININ ON BOUNDS VIOLATION TO
                  DETERMINE IF GOT:  1) TRUE BOUNDS VIOLATION
                  OR  2) AN INDUCED BOUNDS VIO THAT INDICATES
                  THAT THE CONTROL Y TRAP PROCEDURE MAY NOW
                  BE ENTERED).
    OSI           STATE OF THE "ASOFT" PCB BIT WHEN CONTROL Y
                  TRAP WAS ENTERED.  ASOFT = 1 ALLOWS USER SOFT
                  INTERRUPTS AGAINST THE PROCESS.  IT IS SET TO
                  ZERO WHEN THE CONTROL Y HANDLER IS ENTERED.
                  IT IS SET TO ITS PRIOR STATE WHEN THE USER
                  CALLS RESETCONTROL.
 * SET TO COMMAND RECORD LENGTH WHEN COMMAND PENDING
   (I.E. COMMAND ENTERED DURING BREAK OR ENCOUNTERED
   DURING FLUSHING).

** CONTINUE FLAG VALUES
     0 = NO CONTINUE IN EFFECT
     1 = CONTINUE JUST ENCOUNTERED
     2 = CONTINUE IN EFFECT FOR THIS COMMAND

CY FLAG

    PCBXFIXED(56).(1:1)     = SET BY PSEUDOINT WHEN THERE IS A PENDING
                              CONTROL Y WHICH CANNOT BE PROCESSED BECAUSE
                              OF SYSTEM CODE OR PRIVILEGED CODE.  ININ
                              CHECKS THIS BIT ON BOUNDS VIOLATION OR
                              TRACE TRAP.

SI FLAG

    PCBXFIXED(56).(3:1)     = SPECIFIES THE STATE OF THE USER INTERRUPT
                              FLAG WHEN THE CURRENT CONTROL Y WAS PROCESSED.
```

PXFIXED Expansion Bitmap

The PXFIXED bitmap and expansion area is for use in accounting of extra data segments acquired by the process.

PCBX For Core Resident System Process Stacks

```
  |---------------------------------------------| --------------
 0|          DL-a (Seq Rel DL Value)            |0       |
  |---------------------------------------------|        |
 1|          DB-a (Seq Rel DB Value)            |1       |
  |---------------------------------------------|        |
 2|        USER ATTRIBUTES (always -1)          |2       |
  |---------------------------------------------|        |
 3|                    0                        |3       |
  |---------------------------------------------|    PXGLOB
 4|                    0                        |4       |
  |---------------------------------------------|        |
 5|                    0                        |5       |
  |---------------------------------------------|        |
 6|  0    | D| I|             0                 |6       |
  |---------------------------------------------|        |
 7|                    0                        |7       |
  |---------------------------------------------|        |
10|          ACTUAL JOB INPUT LDEV              |8       |
  |---------------------------------------------|        |
11|          ACTUAL JOB OUTPUT LDEV             |9       |
  |---------------------------------------------|        |
12|                    0                        |10      |
  |---------------------------------------------|        |
13|                    0                        |11      |
  |---------------------------------------------| ------------
12|             PXFIXED SIZE (c-b)              |10      |
  |---------------------------------------------|        |
13|             RELATIVE S (S-DB)               |11      |
  |---------------------------------------------|        |
14|             RELATIVE Z (Z-DB)               |12      |
  |---------------------------------------------|        |
15|             INITIAL Q (Q-DB)                |13      |
  |---------------------------------------------|        |
16|             RELATIVE DL (DB-DL)             |14  PXFIXED
  |---------------------------------------------|        |
17|       GENERAL RESOURCE CAPABILITY(-1)       |15      |
  |---------------------------------------------|        |
20|                 RESERVED                    |16      |
  |---------------------------------------------|        |
21|                  0                          |17      |
  |---------------------------------------------|        |
22|                  DL-c                       |18      |
  |---------------------------------------------|        |
23|                  DL-b                       |19      |
  |---------------------------------------------|        
24|                  DL-a                       |20
  |---------------------------------------------|-------------
```

NOTES: 1. There is no PXFILE area.
2. The PXFIXED area is much smaller than a normal PCBX.

Process To Process Communication Table

This table is used as the communication link by which father and son processes communicate with one another via the mailbox scheme. This table contains two words per entry and is indexed by PCB# (entry index 0 is meaningless). Each two word entry of index N essentially relates where, as well as how much, mail may be found for a process N with respect to communications between N and his father process.

```
                    ENTRY FORMAT
                    ------------

                |--------------------|
         word 0 |    WORD COUNT      |
                |--------------------|
         word 1 | MAIL WORD OR DST#  |
                |--------------------|

         where word 0 = the # of mail words to
                        be transferred.
               word 1 = the only word of mail
                        itself if word 0 = 1
                            otherwise
                        it contains the DST# of
                        the extra data segment
                        where "word count" words
                        of mail exist.
```

NOTE: Assume process S is the son of process F. Then the process to process communication table index which will be used for mailbox communication between son S and father F will be that of the son (i.e. S).

Subsystem Reserved DL Area

```
     |-----------------------------------------------|
     |                                               |

                   REMAINING DL AREA

     |                                               |
     |-----------------------------------------------|
DB-12|            RESERVED FOR SORT/MERGE            |DB-10
     |-----------------------------------------------|
DB-11| RESERVED FOR TRACE, TOOLBOX, & BUSINESS BASIC |DB-9
     |-----------------------------------------------|
DB-10|        EXTERNAL PLABEL OF OUTER BLOCK         |DB-8
     |-----------------------------------------------|
 DB-7|      RESERVED FOR TRACE & SYMBOLIC DEBUG      |DB-7
     |-----------------------------------------------|
 DB-6|        DB ADDRESS OF STLT                     |DB-6
     |-----------------------------------------------|
 DB-5|              RESERVED FOR COBOL               |DB-5
     |-----------------------------------------------|
 DB-4|              RESERVED FOR COBOL               |DB-4
     |-----------------------------------------------|
 DB-3|              RESERVED FOR COBOL               |DB-3
     |-----------------------------------------------|
 DB-2|        RESERVED FOR FORMATTER & PASCAL        |DB-2
     |-----------------------------------------------|
 DB-1|              DB ADDRESS OF FLUT               |DB-1
     |-----------------------------------------------|
     |                                               |

                        DB AREA

     |                                               |
     |-----------------------------------------------|
```

FORTRAN Logical Unit Table (FLUT)

The segmenter is responsible for the preparation and initialization of a FORTRAN logical unit table. This is done when a program is prepared if that program contains at least one program unit that references a logical unit. The location of the FLUT is in the secondary DB area and the address of this location is contained in DB-1.

The FLUT is formatted as per the following example:

```
                       |-------|
                  DB-1 |   X   |
                       |-------|

                       |-------|
                  DB+X | 3 | 0 |
                       |---|---|
                       | 4 | 0 |
                       |---|---|
                       | 5 | 0 |
                       |---|---|
                       | 7 | 0 |
                       |---|---|
                       |10 | 0 |
                       |---|---|
                       |255|///|
                       |-------|
                         ^   ^
             -------------|   |-----------------
             |                                 |
         1st BYTE                           2nd BYTE
List of the logical unit numbers     The MPE file number (as returned
referred to in this FORTRAN-         by FOPEN) used in accessing the
produced program.                    file.  Zero if file not open.
(255 terminates).                    Filled in by formatter as each
                                     l.u. is initially referenced.
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
         |                                               |
         |-----------------------------------------------|
         |                                               |
         |-----------------------------------------------|
         |                                               |
         |-----------------------------------------------|
         |                                               |
```

CHAPTER 8 JOB TABLES

Job Tables Overview

Job Master Table (JMAT): One entry per job/session. Contains information needed to get the job/session running. Entry is created at the introduction of job/session.

Job Information Table (JIT): One DST per job/session. Contains information needed by the job/session as it is executing.

Job Process Count Table (JPCNT): One entry per job/session. Entry number used to index into the JIR to lock job resources.

Job Directory Table (JDT): One DST per job/session. Contains the following sub-tables used by descendants of job/session. Must obtain JIR (by using JPCNT index) before accessing JDT. Sub-tables:
1. Data Segment Directory - Directory of DSTs used by job/session
2. Temporary File Directory
3. File Equation Table
4. Line Equation Table
5. Job Control Word Table

Job Cut-off Table (JCUT): Stores total CPU time limit of job/session and accumulates the CPU time that job/session uses.

Ucop Request Queue: A queue of Process Identification Numbers that are terminating.

Job Master Table Structure (JMAT)

```
SIR = 15(10) = %17
DST = 25(10) = %31                                ZEROTH
                                                   ENTRY
                                                     |
                                                     |
                                                     |
     0 1 2 3 4 5 6 7 8 9101112131415                 |
    |-------------------------------|  ----------------------
   0|    MAXSIZE     |    CURSIZE   |0   max JMAT size (words/128)
    |-------------------------------|    current JMAT size (words/128)
   1| VMOUNT INFO    |  ENTRY SIZE  |1   :VMOUNT state saved for WARMSTARTs
    |-------------------------------|    JMAT entry size (38)
   2|         ENTRY POINTER         |2   DB pointer to first entry (38)
    |-------------------------------|
   3|    SCHEDULING HEAD POINTER    |3   DB pointer to word 0 of head
    |-------------------------------|    entry in scheduling queue
   4|    SCHEDULING TAIL POINTER    |4   DB pointer to word 0 of tail
    |-------------------------------|    entry in scheduling queue
   5| TY|        SCOUNTER           |5   next assignable session #, TY=1
   6|                               |6
    |-------------------------------|
   7| TY|        JCOUNTER           |7   next assignable batch #, TY=2
  10|                               |8
    |-------------------------------|
  11|LG|SEC |///////|SFENCE/|JOBFNCE|9   LG=1, logoff in progress
    |-------------------------------|    SEC=0,high;=3,low JOBSECURITY
  12|            SLIMIT             |10  maximum number sessions   C E
    |-------------------------------|                            \ U X
  13|             SNUM              |11  current number sessions | R E
    |-------------------------------|                            | R C
  14|            JLIMIT             |12  maximum # batch jobs    > E U
    |-------------------------------|                            | N T
  15|             JNUM              |13  current # batch jobs    | T I
    |===============================|                            / L N
  16|      JMAT SCHEDHEAD           |14                            Y G
    |-------------------------------|
  17|           WORKAREA            |15  SFENCE is session fence
    |            (23WDS)            |
  20|                               |16


  45|                               |37
    |===============================|  ------------------------------
  46|                               | 38               ^
    |                               |                  |
    |                               |                  |
    |                               |
```

JMAT (Cont.)

```
                                             ENTRY 1

   |                              |           |
   |                              |           |
   |                              |           |
113|                              |75         v
   |==============================|  ------------------------------
   |                              |
   |                              |
   |                              |


   |                              |  LAST
   |                              |  ENTRY
   |==============================|
```

```
SCHEDULING QUEUE
----------------
WAITING SESSIONS
   FIFO WITHIN HIPRI/INPUT PRIORITY
[ERROR JOBS       ]
[  FIFO           ]
 WAITING JOBS
   FIFO WITHIN HIPRI/INPUT PRIORITY
```

Job Master Table Entry (JMAT)

```
                   1 1 1 1 1 1
   0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
   |------------------------------|
  0|  state   :D|I:G:A|U:C: INPRI | 0  state
   |------------------------------|      0 = free entry
  1| ty:   job/session number     | 1    1 = introduced, in
   |------------------------------|          STARTDEVICE
  2| job/session #                | 2  %70 =scheduled in scheduled job queue.
   |------------------------------|
  3|                              | 3  %40 = waiting, job in
  4|        user name             | 4          scheduling queue
  5|                              | 5  %60 = initial, UCOP
  6|                              | 6          has created JSMP
   |------------------------------|      2 = executing, JSMP
  7|                              | 7          finished initial.
 10|        account name          | 8    3 = terminating.
 11|                              | 9    4 = suspended.
 12|                              |10    D = duplicative
   |------------------------------|      I = interactive
 13|                              |11    {G = group password
 14|        job name              |12    {(QUIET mode, if state=2)
 15|                              |13    {A = account password
 16|                              |14    {U = user password
   |------------------------------|      {0 = password validated(STARTDEVICE)
 17|                              |15    {1 = must validate
 20|        group logon name      |16    {    password  (INITJSMP)
 21|                              |17    R = reserved
 22|                              |18
   |------------------------------|      C = JLIST is device
 23|   JIN device                 |19        class index
   |------------------------------|
 24|   JLIST device               |20
   |------------------------------|
 25|   Julian date (CALENDAR)     |21
   |------------------------------|      ty = 1 - session
 26|   time (CLOCK)               |22           2 - job
 27|                              |23
   |------------------------------|
 30| language      :   XPRI       |24
   |------------------------------|
 31|  Main pin                    |25
   |------------------------------|
 32| CPU lim. (0 deflt, -1 no lim.)|26
   |------------------------------|
 33|S|R:N:FT :OUTPRI : NUMCOPIES  |27    ORIGJIN/ORIGJLIST is
   |------------------------------|      used as a scheduling
 34|    ORIGJIN                   |28    link by UCOP (state=
   |------------------------------|      %40).  DB relative ptr.
 35|    ORIGJLIST                 |29    Last entry in list contains zero (0)
   |------------------------------|
 36|    Reserved                  |30
```

JMAT (Cont.)

```
   |-------------------------------|
37|    Reserved                    |31
   |-------------------------------|
40|    Reserved                    |32
   |-------------------------------|
41|    Reserved                    |33
   |-------------------------------|
42|    Reserved                    |34
   |-------------------------------|
43|    Reserved                    |35
   |-------------------------------|
44|    Unused                      |36
   |-------------------------------|
45|    Unused                      |37
   |-------------------------------|

    0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
                       1 1 1 1 1 1
                                        FT = funny terminal
     R = RESTART                             00 - regular term.
     N = SEQUENCED                           01 - regular term.,
     S = ORIGJIN is spooled.                      special logon
                                             10 - APL term.
                                             11 - APL term.
```

Job States

JOB STATES - JMAT ENTRY WORD 0.(0:6)

SHOWJOB - Displays job states by scanning JMAT DST (%31)

LOGON USES ALL STATES EXCEPT "SUSPEND"

STATE NO.	STATE NAME	PROCESS	SEGMENT	PROCEDURE(S)
1	INTRO	DEVREC JSMP SPOOLER	NURSERY	STARTDEVICE ->PUTJMAT ->ALLOCENTRY IN SEGMENT ALLOCUTIL
%70	SCHED	UCOP	JOBSCHED	CXSTSTREAM SCHEDULEDSCHED
%40	WAIT	DEVREC JSMP	NURSERY	STARTDEVICE ->SCHEDULEJOB
		SPOOLER	\ SPOOLING /	SPOOLSTUFFIN ->SCHEDULEJOB
%60	INIT-IALIZAT-ION	UCOP	UCOP	LAUNCHJOB
2	EXEC	JSMP	NURSERY	INITJSMP
3	TERMIN-ATING	JSMP	MORQUE	TERMINATE ->EXPIRE -> CLEANUPJOB
0	FREE ENTRY	JSMP	MORQUE	TERMINATE ->EXPIRE -> CLEANUPJOB ->DEALLOCENTRY IN ALLOCUTIL
4	SUSP	JSMP	OPLOW	CXBREAKJOB

For states INTRO and WAIT,

```
   DEVREC  => logon command originated on terminal or
              other unspooled device.
   SPOOLER => logon command originated on spooled device.
   JSMP    => logon command is the result of the execution of
              a :STREAM command.  (This also includes USER
              processes which have done programmatic :STREAMs.)
```

Job Process Count Table (JPCNT)

(1 Bit Entry/Running Job)

```
                                           MEMORY RESIDENT
                                           ---------------
                                       SYSGLOB BASE = DB+13(%15)
                                       DST = 24(10)
                                       SIR = 13(10)

      0 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5
      |---------------------------------|
   0  | Total Configured number of Jobs |
      |          and Sessions           |
      |---------------------------------|
   1  |  Total number of free entries   |
      |                                 |
      |---------------------------------|
   2  | Bit Map relative index of word  |
      |   containing next free entry    |
      |---------------------------------|
   3  |             unused              |
      |                                 |
      |---------------------------------|
   4  |                                 |   free entry = 1
      |                                 |   allocated entry  = 0
      |            Bit Map              |

                64 words long
      |                                 |
      |---------------------------------|
```

A JPCNT entry must be allocated before the main process can be procreated.

The job SIR (PXGJSIR) = some base+JPCNT index.

NOTE: This table is completely bit oriented with each entry consisting of one bit. Entries are taken from available pool on a "first found" basis. A "1" found in the bit map indicates a free entry. A zero (0) found in the bit map indicates an allocated entry. Word 2 of this table is the index of the word in the Bit Map where the next free entry resides. At system start up, this word is set to zero (0). The Bit Map can be thought of as ranging from 0-63 (64 total words - 1024 entries).

Job Cutoff Table (JCUT)
1 Entry/ CPU-limited Job

```
                                                  MEMORY RESIDENT
                                                  ---------------
                                              SYSGLOB BASE = DB+11(%13)
                                              DST = 36(10);SIR = 14(10)
                                              SYSGLOB + %117 = default
                                               CPU time limit for jobs

        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15

       |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   -------------
       | # OF REAL ENTRIES                                | 0
       |--------------------------------------------------|
       | ENTRY SIZE (3)                                   | 1     HEADER
       |--------------------------------------------------|       ENTRIES
  -----| FREE HEAD                                        | 2
  |    |--------------------------------------------------|        (2)
  | ---| POINTER TO LAST ENTRY (0)                        | 3
  | |  |--------------------------------------------------|
  | |  | UNUSED                                           | 4
  | |  |--------------------------------------------------|
  | |  | UNUSED                                           | 5
  | |  |--------------------------------------------------|   -------------
  | |
  | |
  | |                                                        TYPICAL ENTRY
  | |  |--------------------------------------------------|   -------------
  | |  |                    JCUTCPUL                      |    time limit
  | |  |--------------------------------------------------|    (seconds)
  | |  |                                                  |
  | |  |                    JCUTCPUC                      |    time count
  | |  |                                                  |    (msec)
  | |  |--------------------------------------------------|   -------------
  | |  |                                                  |
  | |
  | |
  | |
  | |  |                                                  |
  | |  |--------------------------------------------------|   -------------
  --|->| POINTER TO NEXT FREE ENTRY (END OF LIST = 0)      |
    |  |--------------------------------------------------|
    |  |                                                  |    FREE ENTRY
    |  |--------------------------------------------------|
    |  |                                                  |
    |  |--------------------------------------------------|   -------------
    -->|                   LAST ENTRY                     |
       |--------------------------------------------------|
       |                                                  |
       |--------------------------------------------------|
       |                                                  |
       |--------------------------------------------------|
```

Job Information Table (JIT)

JIT DST is word 11 (base 10) in PXGLOB

```
                     1 1 1 1 1 1
   0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
  |-------------------------------|
 0| JIT DST                       | 0
  |-------------------------------|
 1|   6      :      not used      | 1
  |-------------------------------|
 2|  pointer to job info        8 | 2
  |-------------------------------|
 3|  pointer to acct info      48 | 3
  |-------------------------------|
 4| pointer to reserved area   59 | 4
  |-------------------------------|
 5|    association table index    | 5
  |-------------------------------|
 6|                             |F| 6    F - Job/Session-wide
  |-------------------------------|           FPMAP option flag
 7|ty :   job number              | 7         (JSFPMAP)
10|                               | 8    ty - 1 = Session
  |-------------------------------|           2 = Job
11|                              7| 9
  |-------------------------------|
12| JITMAXP       :EOF:           |10    JITMAXP - MAXJOBPRI capability
  |-------------------------------|      JITMPN  - Job main PIN.
13| JITMPN                        |11    JITEOF  - used by FCLOSE to tell CI
  |-------------------------------|         that a $STDIN(X) file was closed
14|        DS DATASEG             |12       w/out encountering an EOF.
  |-------------------------------|         (0:1)=$STDIN, (1:1)=$STDINX
15|           JITASEC             |13
  |-------------------------------|
16|        JITGSEC (2 words)      |14
  |        group security         |
  |-------------------------------|
20|        JITHAN (4 words)       |16
  |        account name           |
  |-------------------------------|
24|        JITHGN (4 words)       |20
  |        home group             |
  |-------------------------------|
30|        JITLGN (4 words)       |24
  |        log-on group           |
  |-------------------------------|
  +                               +
   0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
                     1 1 1 1 1 1
```

JIT (Cont.)

```
                     1 1 1 1 1 1
   0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
  |-------------------------------|
34|                               |28
35|            JITUN              |29
36|          user name            |30
37|                               |31
  |-------------------------------|
40| pointer to JITAIP          53 |32
  |-------------------------------|
41|P|M: pointer to JITGIP      55 |33    P - Group's home volume is
  |-------------------------------|          a private volume
42|            LATTR              |34    M - Private volume mounted
43|       local attributes        |35        (i.e. group bound to home
  |-------------------------------|          volume set), JITGIP = 57
44|            PASSF              |36
45|      passed file pointer      |37
  |-------------------------------|
46|            UCAP               |38
47|      user capability *        |39
  |-------------------------------|
50| Reserved for DS'II            |40
  |-------------------------------|
51|///////////////////////////////|41
52|///////////////////////////////|42
  |-------------------------------|
53|      local RIN pointer        |43
  |-------------------------------|
54|                               |44
55|            JITJN              |45
56|           job name            |46
57|                               |47
  |-------------------------------|
  +                               +
   0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
                     1 1 1 1 1 1
```

JIT (Cont.)

```
                      1 1 1 1 1 1
    0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
   |-------------------------------|
 60|                             3 |48   Accounting Info
   |-------------------------------|
 61| JITCREC - # of creations      |49
   |-------------------------------|
 62|          JITCPUC              |50
 63|       cpu milliseconds        |51
   |-------------------------------|
 64|  not used     :     HIPRI     |52     HIPRI - highest job priority
   |-------------------------------|
 65|              0                |53   Account
 66|           JITAIP              |54   Index Pointer
   |-------------------------------|
 67|              0                |55   Group index pointer
 70|           JITGIP              |56   System volume set
   |-------------------------------|
 71|      0       :     MVTABX     |57   Group index pointer
 72|           JITGIP              |58   Mounted private volume set
   |-------------------------------|     MVTABX - Mounted Volume
 73|                             1 |59            Table Index
   |-------------------------------|
 74|                             0 |60
   |-------------------------------|
 75|                               |61
 76|       allow mask              |62
 77|                               |63
100|                               |64
101|                               |65
102|                               |66
   |-------------------------------|
    0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
                      1 1 1 1 1 1
```

Allow Mask Format

The Allow mask for MPE V is expanded to six words. There is a mask in each user's JIT and in the SYSGLOB area. The Allow mask contains enough bits for a one-to-one correspondence to every present OPERATOR type command, or any future OPERATOR command. When a user is ALLOWed any OPERATOR command or ASSOCIATEd to a device (which will use OPERATOR type commands) then the corresponding bit(s) in the mask in that user's JIT for that command is set. If the ALLOW or ASSOCIATE was done on a global scale, then the bit(s) in the mask of the SYSGLOB area is/are updated.

The Following EQUATEs define the mask bit for each operator command.

The first set of commands define the operator commands dealing with devices.

When adding a new command to this set of EQUATEs, be sure to add a corresponding move statement in LOGIMAGE, even if the command will not be logged.

	Word	Bit	#
ABORTIO	0	0	0
ACCEPT	0	1	1
DOWN	0	2	2
GIVE	0	3	3
HEADOFF	0	4	4
HEADON	0	5	5
REFUSE	0	6	6
REPLY	0	7	7
STARTSPOOL	0	8	8
TAKE	0	9	9
UP	0	10	10
MPLINE	0	11	11
DSCONTROL	0	12	12

UPPER LIMIT->DEVICE COMMANDS

	Word	Bit	#
ABORTJOB	0	13	13
ALLOW	0	14	14
ALTFILE	0	15	15
ALTJOB	1	0	16
BREAKJOB	1	1	17
DELETE	1	2	18
DISALLOW	1	3	19
JOBFENCE	1	4	20
LIMIT	1	5	21
STOPSPOOL	1	6	22
SUSPENDSPOOL	1	7	23
OUTFENCE	1	8	24
RECALL	1	9	25
RESUMEJOB	1	10	26
RESUMESPOOL	1	11	27
STREAMS	1	12	28
CONSOLE	1	13	29
WARN	1	14	30
WELCOME	1	15	31
MON	2	0	32
MOFF	2	1	33
VMOUNT	2	2	34
LMOUNT	2	3	35
LDISMOUNT	2	4	36
MRJECONTROL	2	5	37
JOBSECURITY	2	6	38
DOWNLOAD	2	7	39
MIOENABLE	2	8	40
MIODISABLE	2	3	41
LOG	2	10	42

	Word	Bit	#
FOREIGN	2	11	43
INF	2	12	44
SHOWCOM	2	13	45
OPENQ	2	14	46
SHUTQ	2	15	47
DISCRPS	3	2	48

* THE FORMAT FOR UCAP (%46-47) IS AS FOLLOWS:

```
        |-----------------------------------------------|
        | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
        |-----------------------------------------------|
  WORD1 |SM|AM|AL|GL|DI|OP|CV|UV|LG|//|//|NA|NM|CS|NO|SF|
        |-----------------------------------------------|
  WORD2 |                    |BA|IA|PM|     |MR|  |DS|PN|
        |-----------------------------------------------|
```

Job Directory Table (JDT)

```
           |-----------------------|
         0 |   MAX SEG SIZE(WDS)   |             1 entry per job
           |-----------------------|             DST # in word 10
         1 |   POINTER TO JDSD     |             (base 10) of PXGLOB
           |-----------------------|
         2 |   POINTER TO JTFD     |
           |-----------------------|
         3 |   POINTER TO JFEQ     |
           |-----------------------|
         4 |   POINTER TO JLEQ     |
           |-----------------------|
         5 |  POINTER TO JJCW      |
           |-----------------------|
         6 |POINTER TO FREE SPACE  |
           |-----------------------|
                 WORK  AREA
                  15 words
           |-----------------------|
  JDSJNUM  |TY|       NUM          |    job number
           |-----------------------|
           | JSMPIN                |    main process number
           |-----------------------|
                  JOB DATA
  JDSD        SEGMENT DIRECTORY
           |-----------------------|
           |                       |
                JOB TEMPORARY             |----------------------|
  JTFD         FILE DIRECTORY             |ENTRY       |NAME     |
                                          | SIZE (WDS)| SIZE (WDS)|
           |                       |      |----------------------|
           |-----------------------|      |    C1     |    C2    |
           |                       |      |----------------------|
           |                       |   ---
                  JOB FILE             |  |----------------------|
  JFEQ         EQUATION TABLE          |  |    CN     |   (%40)  |
                                       |  |----------------------|
           |-----------------------|   |  |                      |
                  JOB LINE             |  |        ENTRY         |
  JLEQ         EQUATION TABLE          |  |     INFORMATION      |
           |-----------------------|   |  |                      |
           |                       |   |  |----------------------|
              JOB CONTROL WORD         |
              TABLE   (JJCW)          The name is a
           |-----------------------|  concatenation of up to 3 subnames.
           |                       |  Bit 0 of the 1st character of each
           |       FREE SPACE      |  subname is 1.
           |-----------------------|
```

Job Data Segment Directory Entry (In JDT)

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|          4           |           1            |
|-----------------------------------------------|
|                  SEGMENT ID                   |
|-----------------------------------------------|
|         EXTRA DATA SEGMENT DST INDEX          |
|-----------------------------------------------|
|           # OF PROCESSES ACCESSING            |
|-----------------------------------------------|
```

NOTE: A return of %2004 in the INDEX value after using the GETDSEG intrinsic indicates that there is no more room in the Job Directory Table for another job sharable data segment.

Job Temporary File Entry (In JDT)

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| ENTRY SIZE (WORDS)    |   NAME SIZE (WORDS)   |
|-----------------------------------------------|

         NAME-ACTUAL FILE DESIGNATOR            --------- Name is a
                                                   concatenation of up
|-----------------------------------------------|  to three subnames.
|    VOLUME POINTER     |                       |  Bit 0 of the first
|-----------------------|                       |  character of each
|              FILE LABEL POINTER               |  subname is 1.
|-----------------------------------------------|
```

File Equation Table Entry (In JDT)

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| ENTRY SIZE (WORDS)    |   NAME SIZE (WORDS)   |
|-----------------------------------------------|

                      NAME
              (FORMAL DESIGNATOR)
|-----------------------------------------------|
|                     PMASK                     |
|                                               |*
|                                               |
|-----------------------------------------------|
| NAME LENGTH (BYTES)   | DEVICE LENGTH (BYTES) |
|-----------------------------------------------|

             NAME-ACTUAL DESIGNATOR
              (may not be present)
|-----------------------------------------------|

               DEVICE/CLASS NAME
              (may not be present)
|-----------------------------------------------|
|                   FOPTIONS                    |*
|-----------------------------------------------|
|                   AOPTIONS                    |*
|-----------------------------------------------|
|      #BUFFERS         |  INIT ALLOC  |D |T |S |<---disposition
|-----------------------------------------------|      BIT13 DEL
|                  RECORD SIZE                  |      BIT14 TEMP
|-----------------------------------------------|      BIT15 SAVE
| # EXTENTS    |/////////|     BLOCK FACTOR     |
|-----------------------------------------------|
|                     FILE                      |
|-----------------------------------------------|
|                     SIZE                      |
|-----------------------------------------------|
|                   FILE CODE                   |
|-----------------------------------------------|
|   OUTPRI  |     NUMCOPIES       |             |
|-----------------------------------------------|
|   REF COUNT      |     # OF USER LABELS       |
|-----------------------------------------------|
|      LANG (Native Language Support)           |
|-----------------------------------------------|
|          LENGTH  FORMS*/LABEL*                |
|-----------------------------------------------|
|                 FORMS/LABEL                   |
                     ARRAY

|-----------------------------------------------|
```

Job Line Equation Entry

```
  |----------------------------------------------|
  |  ENTRY SIZE (WORDS)  |   DESIG. SIZE (WORDS) |
  |----------------------------------------------|
  |                   FORMAL                     |
                  LINE DESIGNATOR
                   (1-4 WORDS)
  |----------------------------------------------|
 0|                   PMASK1                     |0
  |----------------------------------------------|
 1| REF CNT        S|P |         PMASK2          |1 P=FLAG
  |----------------------------------------------|
 2|     NAME LENGTH      |        DEV LENGTH     |2
  |----------------------------------------------|
 3|                                              |3
  |                                              |
 4|                    NAME                      |4
  |                                              |
 5|     ( END OF LEQ ENTRY IF NON-BLANK )        |5
  |                                              |
 6|                                              |6
  |----------------------------------------------|
 7|                                              |
  |                                              |
10|                                              |8
  |                   DEVICE                     |
11|                                              |9
  |                                              |
12|                                              |10
  |----------------------------------------------|
13|                   PMASK3                     |11
  |----------------------------------------------|
14| DRIVER NAME LENGTH    |                      |12
  |----------------------------------------------|
15|                                              |13
  |                                              |
16|                                              |14
  |                DRIVER NAME                   |
17|                                              |15
  |                                              |
20|                                              |16
  |----------------------------------------------|
21|                 LIST PNTR                    |17
  |----------------------------------------------|
22|                 COPTIONS                     |18
  |----------------------------------------------|
23|                 AOPTIONS                     |19
  |----------------------------------------------|
24|                 DOPTIONS                     |20
  |----------------------------------------------|
```

JLEQ Entry (Cont.)

```
  |----------------------------------------------|
25|              NUMBER OF BUFFERS               |21
  |----------------------------------------------|
26|             BUFFER SIZE IN WORDS             |22
  |----------------------------------------------|
27|              INSPEED (2 words)               |23
  |----------------------------------------------|
31|              OUTSPEED (2 words)              |25
  |----------------------------------------------|
33|                 POLL REPEAT                  |27
  |----------------------------------------------|
34|                 POLL DELAY                   |28
  |----------------------------------------------|
35|                C TRACE INFO                  |29
  |----------------------------------------------|
36|                LOCAL ID PNTR                 |30 \
  |----------------------------------------------|    |
37|                REMOTE ID PNTR                |31  |
  |----------------------------------------------|    |
40|                SUPLIST PNTR                  |32  | REL TO ORIG
  |----------------------------------------------| .  | OF LEQ ENTRY
41|               PHONE LIST PNTR                |33  |
  |----------------------------------------------|    |
42|                POLLIST PNTR                  |34  |
  |----------------------------------------------|    |
43|               MISC ARRAY PNTR                |35 /
  |----------------------------------------------|
```

Job Control Word Table (JJCW)

```
|--------------------------------|   Name may be any alpha-
| NAME SIZE (BYTES)|             |   numeric string, begin-
|------------------|             |   ning with an alpha,
|                                |   between 1 and 255 char-
                                     acters long.
              NAME
|                                |   TY  00 = OK
|--------------------------------|       01 = WARN
| TY |        MODIFIER           |       10 = FATAL
|--------------------------------|       11 = SYSTEM

MODIFIER = VALUE FROM 0 TO %377777
```

Aoptions and Foptions Word Breakdown

```
OPTION WORD 2            OPTION WORD 1
 (AOPTIONS)               (FOPTIONS)

 0|---|                   0|---|
  | 0 |                    | 0 |
  |   |                    |   |
  | 0 |                    | 0 |
  |   |                    |---|
  | 0 |                   2|   |
  |---|                    |   |file type
 3|   |copy               3|   |
  |---|                    |---|
 4|   |no-wait             | 0 |
  |---|                    |---|
 5|   |                   5| 0 |disallow files
  |   |multi-              |---|
 6|   |access             6|   |labelled tape
  |---|                    |---|carriage
 7|   |inhibit buff.      7|   |control
  |---|                    |---|
 8|   |                   8|   |
  |   |exclusive           |   |record format
 9|   |                   9|   |
  |---|                    |---|
10|   |dynamic locking   10|   |
  |---|multi-              |   |default
11|   |record              |   |designator
  |---|                    |   |
12|   |                  12|   |
  |   |                    |---|
  |   |access type       13|   |ascii/binary
  |   |                    |---|
  |   |                  14|   |
  |   |                    |   |domain
15|   |                  15|   |
  |---|                    |---|
```

PMASK Word Breakdown

```
                  --------- PMASK WORD 2
                  |   ----- PMASK WORD 1
                  |   |
                |---|---|0
FILE TYPE       |   |   |BLOCK FACTOR
                |---|---|
LABELLED TAPE   |   |   |RECSIZE
                |---|---|
FRMS MESSAGE    |   |   |DISPOSITION
                |---|---|
USER LABELS     |   |   |NUMBUFFERS
                |---|---|
LANG            |   |   |INHIBIT BUFFERING
                |---|---|
VTERM           |   |   |EXCLUSIVE
                |---|---|
POINTER ENTRY   |   |   |MULTI-RECORD
                |---|---|
DYN.LOCKING     |   |   |ACCESS TYPE
                |---|---|
WAIT,NOWAIT     |   |   |COPY,NOCOPY
                |---|---|
MULTI ACCESS    |   |   |CARRIAGE CONTROL
                |---|---|
NUMCOP          |   |   |RECORD FORMAT
                |---|---|
OUTPRI          |   |   |DEFAULT DESIGNATOR
                |---|---|
FILECODE        |   |   |ASCII/BINARY
                |---|---|
FILESIZE        |   |   |DOMAIN
                |---|---|
NUMEXTS         |   |   |DEVICE
                |---|---|
INIT ALLOC      |   |   |NAME
                |---|---|
                         15

      1->info present
      0->info absent
```

UCOP Request Queue (DST#9)

```
       |-------------------------------------------|
      0|            MAXM REQ ENTRIES N/2           |
       |-------------------------------------------|
      1| TABLE RELATIVE POINTER TO NEXT AVAIL ENTRY |------
       |-------------------------------------------|      |
      2|   TABLE RELATIVE POINTER TO NEXT REQUEST  |---   |
       |-------------------------------------------|  |   |
      3|                     0                     |  |   |
    -- |-------------------------------------------|  |   |
   |   |                                           |  |   |
   |   |                                           |  |   |
   |   |                                           |  |   |
   |   |                                           |  |   |
   |   |-------------------------------------------|  |   |
   |   |                  REQ 1                    |<--   |
   |   |-------------------------------------------|      |
   |   |                  REQ 2                    |      |
   |   |-------------------------------------------|      |
   |   |                    .                      |      |
   |   |                    .                      |      |
  N    |                    .                      |      |
  WRDS|                     .                      |      |
   |   |                    .                      |      |
   |   |-------------------------------------------|      |
   |   |                  REQ N                    |      |
   |   |-------------------------------------------|      |
   |   |                                           |      |
   |   |                                           |<-----
   |   |                                           |
   |   |                                           |
   |   |                                           |
    -- |-------------------------------------------|
```

UCOP Entry Format

Each entry is
2 words long

```
 0                                      12-15
|---------------------------------------------|
|//////////////////////////////////////   2   |
|---------------------------------------------|  2  process deletion
| PIN                                         |
|---------------------------------------------|
```

CHAPTER 9 RELOCATABLE OBJECT CODE

USL Files Introduction

- * USL record length 128 words always.
- * Layout of doubleword disc addresses

```
|-----------------------------------------------|
|                          |      WORD #        |
|      25-BIT RECORD #     |   WITHIN RECORD    |
|-----------------------------------------------|
0                         24 25                 31
```

- * Hash links join all entries with the same hash key regardless of type.
- * Linear lists terminate with a zero link
- * Circular lists containing only the list head point directly to themselves.
- * Single-word disc addresses

```
|-----------------------------------------------|
|                          |      WORD #        |
|      9-BIT RECORD #      |   WITHIN RECORD    |
|-----------------------------------------------|
0                          8 9                  15
```

Uninitialized fields are reserved for future use and should be set to zero.

Record 0 and Overall USL File Format

```
   -----------                          NOTE:
 0|   LID    | 0   LOADER ID           S.A. = Starting Address
  |----------|
 1|   NE     | 1   NR. DIRECTORY ENTRIES
  |----------|
 2|   DL     | 2   DIR. LENGTH
  |----------|
 3|  SUMDG   | 3   TOTAL DIR. GARBAGE
  |----------|
 4|   NDG    | 4   NR. DIR. GARB. ENTRIES
  |----------|
 5|  SABDL   | 5   S.A. BLOCK DATA LIST
  |----------|
 6|  SAIPL   | 6   S.A. INTERRUPT PROC. LIST
  |----------|
 7|  SASL    | 7   S.A. SEGMENT LIST
  |----------|
  |----------|
10|   FL     | 8   FILE LENGTH
11|          | 9
```

USL File Format (cont.)

```
    |---------|
 12|  SAAD   | 10  S.A. AVAIL. DIR.
    |---------|
 13|   ADL   | 11  AVAIL. DIR. LENGTH
    |---------|
 14|   SAI   | 12  S.A. INFO BLOCK
 15|         | 13
    |---------|
 16|   IL    | 14  INFO BLOCK LENGTH
 17|         | 15
    |---------|
 20|  SAAI   | 16  S.A. AVAIL. INFO
 21|         | 17
    |---------|
 22|   AIL   | 18  AVAIL. INFO LENGTH
 23|         | 19
    |---------|
 24| TOTAL   | 20  TOTAL INFO GARBAGE
 25| I.G.    | 21
    |---------|
 26|  NIG    | 22  NR. INFO  GARB. ENTRIES
    |---------|
 27|         | 23
    |         |
 30|         | 24
    |         |
 31|         | 25
    |         |
 32|         | 26
    |         |
 33|         | 27
    |         |
 34|         | 28
    |         |
 35|         | 29
    |         |
 36|         | 30
    |         |
 37|         | 31
    |         |
 40|         | 32
    |---------|
 41|   HL    | 33  HASH LINKS
    |    0    |
    |---------|
    |    .    |
    |    .    |
    |---------|
177|   HL    | 127
    |   94    |
    -----------
```

USL Files General Information (cont.)

```
                  0-------------0      ^
                   |            |      |
                   | RECORD 0   |      |
                   |            |      |
                   |            |      |
                177-------------127    |
                                       |
 ---            200-------------128    |
  |                |            |      |
  |                | DIRECTORY  |     32K
  |                |            |     MAX
  DL               |  ENTRIES   |      |
  |                |            |      |
  |                |            |      |
 ---               -------------       |
  |                      |             |
 ---               -------------       |
  |            SAAD |           |      |
  |                 | AVAILABLE |      |
 ADL                | DIRECTORY |      |
  |                 |           |      |
 ---               -------------       |

               ---------------------------

 ---           SAI* -------------
  |                 |           |
  |                 |   INFO    |
  IL                | (HEADERS) |
  |                 |  (CODE)   |
  |                 |           |
  |                 |           |
 ---                -------------
                          |
 ---           SAAI -------------
  |                 |           |
 AIL                | AVAILABLE |
  |                 |   INFO    |
  |                 |           |
 ---           FL-1 -------------
*SAI MUST BE ON A RECORD BOUNDARY
```

NOTE: ALL ADDRESSES IN RECORD 0 ARE WORD ADDRESSES.

USL Files General Information (cont.)

```
SASL--->----------         ----------         ----------
     |   SEGL |  -------->|   SEGL  |--------->|    0   |
     |        |           |         |          |        |
     | SEG.A  |           | SEG.K   |          | SEG.B  |
     |        |           |         |          |        |
  ---|        |---      --|SUBL     |<--    ---|        |---
     ----------   |    |  -----------  |   |   ----------
                  |    |               |   |
                  |    |               |   |
-------------------------          -------------------------
|                                                          |
|                                                          |
|                                                          |
|                                                          |
|     ----------         -----------       ----------      |
|------>|  SUBL |--------->|     SUBL|--------->|  SUBL|------|
      |        |         |         |         |        |
      | PROC.C |         | PROC.A  |         | MAIN   |
      |        |         |         |         |        |
   ---|        |---    ---|SECL     |<--   ---|        |---
      ----------   |   |  -----------  |   |  ----------
                   |   |               |   |
                   |   |               |   |
--------------------------          ---------------------------
|                                                            |
|               CIRCULAR LINK POINTS TO ITSELF               |
|               IF LIST IS EMPTY                             |
|                                                            |
|                                                            |
|     ----------        ----------        ---------          |
------->|  SECL |--------->|    SECL|--------->|   SECL|----------
      |        |         |        |        |       |
      |PROC.A  |         |PROC.A  |        |PROC.A |
      |      3 |         |      1 |        |     5 |
      |        |         |        |        |       |
      ----------         ----------        ---------


A \                                     PROC C \
K  >SEGMENT NAME ENTRIES                PROC A  >SUBPROGRAM
B /                                     MAIN    / ENTRIES

          A  \
           3 |
          A  |
           1  } SECONDARY ENTRY POINT ENTRIES
          A  |
           5 /
```

Data Descriptors, Passed Parameters

```
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
---|-|-|-|-|-|-|-|-|-|--|--|--|--|--|----
| MODE   | STRUCTURE |       TYPE        |
-----------------------------------------
```

TYPE	WORDS	CODE
NULL		0
LOGICAL	1	1
INTEGER	1	2
BYTE	1/2	3
REAL	2	4
DOUBLE	2	5
LONG	3	6
COMPLEX	4	7
LABEL (SPL)		10
CHARACTER (STRING)	N/2	11
LABEL (FORTRAN)		12
UNIVERSAL (MATCHES ANY TYPE)		13

STRUCTURE	
SIMPLE VARIABLE	0
POINTER	1
ARRAY	2
PROCEDURE	3

MODE	
NULL	0
VALUE	1
REFERENCE	2
NAME	3

NOTE: A descriptor of 0 results in an automatic match.

Pascal

Pascal sets the high order bit in the parameter type descriptor when it is generating hashed values. The remaining 15 bits are based on a hash of the types of the parameter. Only the Pascal compiler can compute the value, and the SEGMENTER must match the whole 16 bit value.

Entry Type 0

GARBAGE

```
  0  1                   10 11     15
-------------------------------------
|////|        NW          |   0    |
-------------------------------------
|                                   |
|                                   |
|            GARBAGE                |
|                                   |
|                                   |
-------------------------------------
```

NW - Number of words in this block

Entry Type 1

SEGMENT NAME

```
 0  1              7 8   10 11    15
-------------------------------------
|///|        NW           |   1    |
-------------------------------------
|                H L                |
-------------------------------------
|A |////////| NC          CHAR1     |
-------------------------------------
|               .                   |
|   (VARIABLE # CHAR. SEE NC)       |
|               .                   |
|               .                   |
-------------------------------------
|   CHAR. NC     |/////////////////|
-------------------------------------
|              SEGL                 |
-------------------------------------
| L |          SUBL                 |
-------------------------------------
```

NW - Number of words in entry block

HL - Hash link - points to next entry having the same hash code

A - Activity bit
0 if active
1 if inactive
(initialize to 0)

Note: An inactive segment implies that all entry points are inactive

NC - Number of characters in name. Max is 16

CHAR. 1 - First character in variable field
CHAR. NC - Last character in variable field
SEGL - Segment link - points to next segment name entry
SUBL - Subprogram link - points to next entry having the same segment name
L - Last entry in list
0 if not last
1 if last

Clarification Notes on Entry Types 2 and 4 With Respect to SPL and FORTRAN

*ENTRY TYPE 2 SPL O.B.	**ENTRY TYPE 4 SPL PROC	*ENTRY TYPE 2 FORTRAN MAIN	**ENTRY TYPE 4 FORTRAN SUB.
TPDB	0	0	0
TSDB[1,5]	TSDB[1]	TSDB[1,2,3,4]	TSDB[1,2,3,4]
NWPUST	NWPUST	NWPUST	NWPUST
NWSDB[5]	NWO	NWD	NWD

WHERE:
TPDB = Total primary DB length in words
TSDB = Total secondary DB length in words
NWPUST = Number of words in "TRACE" array
NWSDB = Number of words in secondary DB array
NWO = Number of words in own array
NWD = Number of words in data array

Notes:
1. Does not include the length of the STLT
2. Does not include the length of the FLUT
3. Does not include the length of any common array
4. Includes the length of any DB-allocated format array
5. Are not necessarily equal

In general TPDB and TSDB are summations of storage allocated in the global area of the program's data segment. They are not, however, complete since the compilers are not aware of all storage actually allocated! The STLT and FLUT are examples of this since these tables are constructed by the segmenter. Common arrays also present a problem since their inclusion in TPDB and TSDB might cause their storage requirements to be counted more than once.

Entry Type 2

OUTER BLOCK

```
 0   1  2  3  4 5 6 7  8     10 11    15
----------------------------------------
|//|           NW              |   2   |
----------------------------------------
|                 HL                   |
----------------------------------------
| A | C | I |///| NC |     CHAR 1      |
----------------------------------------
|                  .                   |
|       (VARIABLE # CHAR.SEE NC)       |
|                  .                   |
----------------------------------------
|     CHAR NC       |//////////////////|
----------------------------------------
| L |              SUBL                |
----------------------------------------
| L |              SECL                |
----------------------------------------
|                  SSA                 |
----------------------------------------
|                  SAC                 |
|     RELATIVE TO SAI (SEE RECORD 0)   |
|                                      |
----------------------------------------
| F | W |          NWC                 |
----------------------------------------
|                  SE                  |
----------------------------------------
|                 TPDB                 |
----------------------------------------
|                 TSDB                 |
----------------------------------------
|                NWPUST                |
----------------------------------------
|               NWD/NWSDB              |
----------------------------------------
| T |              NH                  |
----------------------------------------
|                  SAH                 |
|     RELATIVE TO SAI (SEE RECORD 0)   |
----------------------------------------
|                  HDW                 |
----------------------------------------
```

Entry Type 2 (cont.)

```
-----------------------------------------
|                  .                    |
|                  .                    |
|                  .                    |
-----------------------------------------
|                 HDW                   |
-----------------------------------------
|                  .                    |
|                  .                    |
|                  .                    |
-----------------------------------------
| T |             NH                    |
-----------------------------------------
|                                       |
|                 SAH                   |
|                                       |
-----------------------------------------
|                 HDW                   |
-----------------------------------------
|                  .                    |
|                  .                    |
|                  .                    |
-----------------------------------------
|                 HDW                   |
-----------------------------------------
```

NW - Number of words in entry block.

HL - Hash link - points to next entry with
     same hash code.

A - Activity bit. 0 if active, 1 if inactive
    outer block.

C - Callability bit set if entry point is
    uncallable.

I - Privilege mode bit - set if program unit is
    to be executed in Privilege mode..

NC - Number of characters in name. Max is 16.

CHAR. 1 - First character in variable field.

CHAR. NC - Last character in variable field.

L - Last entry in list.
    0 if not last
    1 if last

Entry Type 2 (cont.)

SUBL - Subprogram link - points to next entry
       Entry having the same segment name.

SECL - Secondary entry point list link.

SSA - Program unit starting PB address.

SAC - Starting 8FILE9 address of code
      module

F - Set if fatal error

W - Set if nonfatal error

NWC - Number of words in code module.

SE - Stack size estimate

TPDB - Total number of words of primary
       DB to be allocated

TSDB - Total number of words of secondary
       DB to be allocated.

NWPUST - Number of words in trace array
         (PUST)

NWD - Number of words in data array
      (FORTRAN)

NWSDB - Number of words in secondary
        DB array (SPL)

T - Terminating bit - set if last set of
    headers in entry

NH - Number of headers

SAH - Starting address of header (relative
      to SAI)

HDW - Header (pointer)

Entry Type 3

OUTER BLOCK - SECONDARY ENTRY POINT

```
 0   1  2  3  4 5 6 7  8     10 11    15
----------------------------------------
|//|             NW             |  3   |
---------------------------------------|
|                HL                    |
----------------------------------------
| A | C |//|//| NC  |     CHAR 1       |
----------------------------------------
|                 .                    |
|     (VARIABLE # CHAR.SEE NC)         |
|                 .                    |
----------------------------------------
|    CHAR NC      |////////////////////|
----------------------------------------
| L |             SECL                 |
----------------------------------------
|                 SSA                  |
----------------------------------------
```

Entry Type 4

PROCEDURE

```
 0   1  2   3 4567 8                  10 11        15
---|--|--|--|----|-------------------|------------
|//|              NW                  |     4     |
-------------------------------------------------
|                     HL                        |
-------------------------------------------------
|A | C| I| H| NC |          CHAR.1              |
-------------------------------------------------
|                     .                         |
|          (VARIABLE # CHAR. SEE NC)            |
|                     .                         |
-------------------------------------------------
| CHAR.NC       |///////////////////////////////|
-------------------------------------------------
|L |              SUBL                          |
-------------------------------------------------
|L |              SECL                          |
-------------------------------------------------
|                 SSA                           |
-------------------------------------------------
```

Entry Type 4 (cont.)

```
-------------------------------------------------
|                                               |
|                    SAC                        |
|                                               |
-------------------------------------------------
|F | W|              NWC                        |
-------------------------------------------------
|                     SE                        |
-------------------------------------------------
|                    TPDB                       |
-------------------------------------------------
|                    TSDB                       |
-------------------------------------------------
|                   NWPUST                      |
-------------------------------------------------
|                   NWD/NWO                     |
-------------------------------------------------
|  P  |     NP    |          CN                 |
-------------------------------------------------
|                     TN                        |
-------------------------------------------------
|                   PARM.1                      |
-------------------------------------------------
|                      .                        |
|      (VARIABLE # OF PARMS. SEE CN)            |
|                      .                        |
-------------------------------------------------
|                   PARM. NP                    |
-------------------------------------------------
| T|                  NH                        |
-------------------------------------------------
|                                               |
|                     SAH                       |
|                                               |
-------------------------------------------------
|                     HDW                       |
-------------------------------------------------
|                      .                        |
|                      .                        |
|                      .                        |
-------------------------------------------------
|                     HDW                       |
-------------------------------------------------
|                      .                        |
|                      .                        |
|                      .                        |
-------------------------------------------------
|                     ETC                       |
-------------------------------------------------
```

Entry Type 4 (cont.)

NW - Number of words in entry block
HL - Hash link - points to next entry with same hash code
A - Activity bit. 0 if active, 1 if inactive entry point
C - Callability bit set if entry point is uncallable
I - Privilege mode bit. Set if procedure is to be executed in privilege mode.
H - Hidden entry point. Set if entry point will not be in
    library directory.
NC - Number of characters in name. Max is 16.
CHAR1 - First character in variable field.
CHAR NC - Last character in variable field.
L - Last entry in list
    0 if not last
    1 if last
SUBL - Subprogram link. Points to next entry having the same segment
    Name
SECL - Secondary entry point list link.
SSA - Unit starting PB address
SAC - Starting (file) address of code module
F - Set if fatal error
W - Set if nonfatal error
NWC - Number of words in code module
SE - Stack size estimate
TPDB - Total number of words of primary DB to be allocated.
TSDB - Total number of words of secondary DB to be allocated.
NWPUST - Number of words in trace array (PUST)
NWD - Number of words in data array (FORTRAN)
NWO - Number of words in own array (SPL)
P - Parameter checker
    00 no checking. (Implies NP undefined, FN and PARM's absent)
    01 check procedure type. (Implies NP is undefined and PARM's
        absent)
    10 check procedure type and number of PARM's (implies PARM's
        absent)
    11 check procedure type, number of PARM 's and type of each PARM.
NP - Number of PARM's
CN - Character count of PARM's
TN - Terminating bit. Set if last set of headers in entry.
NH - Number of headers
SAH - Starting address of header
HDW - Header (pointer)

Entry Type 5

PROCEDURE - SECONDARY ENTRY POINT

```
 0  1  2  3  4 5 6 7  8    10 11      15
|--|-------------------|--------|-------|
|//|          NW                |   5   |
|---------------------------------------|
|                    HL                 |
|---------------------------------------|
| A| C |//|H |  NC   |     CHAR. 1      |
|---------------------------------------|
|                  .                    |
|  (VARIABLE #CHAR. SEE NC)             |
|                  .                    |
|---------------------------------------|
|    CHAR. NC          |////////////////|
|---------------------------------------|
|L |          SECL                      |
|---------------------------------------|
|             SSA                       |
|---------------------------------------|
```

NW - Number of words in entry block

HL - Hash link - points to next entry with
    same hash code

A - Activity bit. 0 if active, 1 if inactive
    entry point

C - Callability bit set if entry point is
    uncallable.

H - Hidden entry point set if entry point
    will not be in library directory

NC - number of characters in name, max
    is 16

CHAR 1 - First character in variable field.

L - Last entry in list
    0 if not last
    1 if last

SECL - Secondary entry point list link

SSA - Unit starting PB' address

Entry Type 6

INTERRUPT PROCEDURE

```
| 0|1 |2 |3 |4567|8      10|11    15|
-------------------------------------
|//|      NW              |   6    |
-------------------------------------
|             HL                    |
|                                   |
-------------------------------------
|A | IT  |//| NC |       CHAR.1     |
-------------------------------------
|                .                  |
|      (VARIABLE # CHAR. SEE NC)    |
|                .                  |
-------------------------------------
|A | IT  |//| NC |       CHAR.1     |
-------------------------------------
|                .                  |
|      (VARIABLE # CHAR. SEE NC)    |
|                .                  |
-------------------------------------
| CHAR. NC      |///////////////////|
-------------------------------------
|              IPL                  |
-------------------------------------
|              DBS                  |
-------------------------------------
|              SSA                  |
-------------------------------------
|                                   |
|              SAC                  |
|                                   |
-------------------------------------
|F | W|        NWC                  |
-------------------------------------
|T |            NH                  |
-------------------------------------
|                                   |
|              SAH                  |
|                                   |
-------------------------------------
|              HDW                  |
-------------------------------------
|               .                   |
|               .                   |
|               .                   |
-------------------------------------
|              HDW                  |
-------------------------------------
```

Entry Type 6 (cont.)

NW - Number of words in entry block

HL - Hash link. Points to next entry
    with same hash code

A - Activity bit. 0 if active, 1 if
    inactive entry.

IT - Interrupt procedure type number

NC - Number of characters in name (maximum is 16)

CHAR 1 - First character in variable
    field.

CHAR NC- Last Character in variable field

IPL - Interrupt procedure link

DBS - Number of words of DB storage
    required.

SSA - Unit starting PB' address

SAC - Starting (file) address of code
    module.

F - Set if fatal error

W - Set if nonfatal error

NWC - Number of words in code module

T - Terminating bit. Set if last set
    of headers in entry.

NH - Number of headers

SAH - Starting address of header.

HDW - Header (pointer)

Entry Type 7

BLOCK DATA

```
| 0 | 1 | 2 | 3 |4567|8        10|11        15|
----|---|---|---|----|-----------|------------
|///|          NW               |     7      |
----------------------------------------------
|                   HL                       |
----------------------------------------------
| A | F | W |///| NC |         CHAR.1        |
----------------------------------------------
|                    .                       |
|             BLOCK DATA NAME                |
|                    .                       |
----------------------------------------------
|    CHAR.NC          |//////////////////////|
----------------------------------------------
|                   BDL                      |
----------------------------------------------
|                   CAL                      |
----------------------------------------------
|///////////////| NC |        CHAR.1         |
----------------------------------------------
|                    .                       |
|            COMMON ARRAY NAME               |
|                    .                       |
----------------------------------------------
|     CHAR.NC          |/////////////////////|
----------------------------------------------
| T |                NH                      |
----------------------------------------------
|                   SAH                      |
|                                            |
----------------------------------------------
|                   HDW                      |
----------------------------------------------
|                    .                       |
|                    .                       |
|                    .                       |
----------------------------------------------
|                   HDW                      |
---------------------------------------------|
|                    .                       |
|                    .                       |
|                    .                       |
----------------------------------------------
```

Entry Type 7 (cont.)

```
----------------------------------------------
|                   CAL                      |
----------------------------------------------
|///////////////| NC |        CHAR.1         |
----------------------------------------------
|                    .                       |
|            COMMON ARRAY NAME               |
|                    .                       |
----------------------------------------------
|     CHAR.NC          |/////////////////////|
----------------------------------------------
| T |                NH                      |
----------------------------------------------
|                   SAH                      |
|                                            |
----------------------------------------------
|                   HDW                      |
----------------------------------------------
|                   ETC                      |
|                                            |
```

NW   Number of words in block

HL   - Hash link. Points to next entry with
        same hash code.

A   - Activity bit. 0 if active, 1 if inactive
        block.

F   - Set if fatal error.

W   - Set if nonfatal error.

CHAR 1- First character in variable field.

CHAR NC-Last character in variable field.

BDL   - Block data link

CAL   - Common array length

T   - Terminating bit. Set if last set of
        headers in entry.

NH   - Number of headers.

SAH   - Starting address of headers.

HDW   - Header (pointer)

Entry Type 8

PROCEDURE - SECONDARY ENTRY POINT

```
  0  1  2  3  4 5 6 7  8     10 11      15
|---|--|--|--|---------|---------|----------|
|///|         NW               |    8      |
|-------------------------------------------|
|                 HL                        |
|-------------------------------------------|
| A | C|//| H|   NC   |     CHAR. 1         |
|-------------------------------------------|
|                  .                        |
|       (VARIABLE #CHAR. SEE NC)            |
|                  .                        |
|-------------------------------------------|
|    CHAR. NC        |//////////////////////|
|-------------------------------------------|
| L |               SECL                    |
|-------------------------------------------|
|                   SSA                     |
|-|-------|------|--------|--------|------|
|  P |      NP     |          CH            |
|-------------------------------------------|
|                   TN                      |
|-------------------------------------------|
|                 PARM. 1                   |
|-------------------------------------------|
|                   .                       |
|                   .                       |
|                   .                       |
|-------------------------------------------|
|                PARM. NP                   |
---------------------------------------------
```

NW - NUMBER OF WORDS IN ENTRY BLOCK

HL - HASH LINK - POINTS TO NEXT ENTRY
     WITH SAME HASH CODE

A - ACTIVITY BIT. 0 IF ACTIVE, 1 IF INACTIVE
    ENTRY

C - CALLABILITY BIT SET IF ENTRY POINT IS
    UNCALLABLE

H - HIDDEN ENTRY POINT. SET IF ENTRY
    POINT WILL NOT BE IN LIBRARY
    DIRECTORY

NC - NUMBER OF CHARACTERS IN NAME. MAX
     IS 16

Entry Type 8 (cont.)

CHAR 1 - FIRST CHARACTER IN VARIABLE LIST

CHAR NC - LAST CHARACTER IN VARIABLE
          LIST

L - LAST ENTRY IN LIST
    0 IF NOT LAST
    1 IF LAST

SECL - SECONDARY ENTRY POINT LIST LINK

SSA - UNIT STARTING PB' ADDRESS

P - PARM CHECKER
    00 NO CHECKING (IMPLIES NP UNDEFINED,
       TN AND PARMS ABSENT)
    01 CHECK PROCEDURE TYPE (IMPLIES NP
       IS UNDEFINED AND PARMS ABSENT)
    10 CHECK PROCEDURE TYPE AND NUMBER
       OF PARMS. (IMPLIES PARMS ABSENT)
    11 CHECK PROCEDURE TYPE, NUMBER OF
       PARMS AND TYPE OF PARM.

NP - NUMBER OF PARMS

CH - CHARACTER COUNT OF PARMS

TN - PROCEDURE TYPE

Entry Header Format

```
              --------------
SAH ------->|   HEADER    |
             |------------|
             |      .      |
             |      .      |
             |      .      |
             |------------|
             |   HEADER    |
              --------------
              --------------
SAH ------->|   HEADER    |
             |------------|
             |      .      |
             |      .      |
             |      .      |
             |------------|
SAC ------->|    CODE     |
             |------------|
             |      .      |
             |             |
             |      .      |
             |------------|
             |   HEADER    |
              --------------


              --------------
SAH ------->|   HEADER    |
             |------------|
             |      .      |
             |      .      |
             |      .      |
             |------------|
             |   HEADER    |
              --------------
```

EACH ENTRY (EXCEPT SECONDARY ENTRY POINT ENTRIES) MAY DESCRIBE N> 0 SETS OF HEADERS. THE HEADERS IN EACH SET MUST BE CONTINUOUS AND IN THE SAME ORDER AS THE HOW LIST DESCRIBING THE SET.

THE CODE MODULE MAY BE PLACED IN ANY POSITION IN A HEADER SET. NOTE THAT IF THE CODE MODULE IS AT THE BEGINNING OF A SET, SAC = SAH.

IF THE ENTRY HAS NO HEADER SET, THEN NH, SAH SEQUENCE IS ABSENT.

Header Type 0

GARBAGE

```
  0  1                          10 11    15
|---|----------------------------|--------|
|///|            NW              |   0    |
|-----------------------------------------|
|                                         |
|               GARBAGE                   |
|                                         |
-------------------------------------------
```

Header Type 1

PCALs

```
  0  1  2  3  4 5 6 7  8      10 11      15
|--|--|--|--|--------|---------|---------|
|//|                            |    1     |
|-----------------------------------------|
|                   PBA                   |
|-----------------------------------------|
|///////////|   NC    |     CHAR. 1       |
|-----------------------------------------|
|                    .                    |
|                    .                    |
|                    .                    |
|-----------------------------------------|
|      CHAR. NC      |////////////////////|
|-----------------------------------------|
|  P  |     NP      |         CN          |
|-----------------------------------------|
|                    TN                   |
|-----------------------------------------|
|                 PARM. 1                 |
|-----------------------------------------|
|                    .                    |
|                    .                    |
|                    .                    |
|-----------------------------------------|
|                 PARM. NP                |
-------------------------------------------
```

PBA - PB' ADDRESS OF LINKED LIST OF PCAL INSTRUCTIONS TO BE REPAIRED - LOWER 14 BITS USED AS NEGATIVE DISP. - BIT 0 SET MEANS THAT WORD IS NOT A PCAL INSTRUCTION BUT A POINTER TO A SST LABEL OF ''EXTERNAL'' FORMAT - A LINK OF 0 TERMINATES THE LIST - BIT 1 SET MEANS THAT THE WORD IS TO BE

INITIALIZED WITH THE PB ADDRESS OF THE PROCEDURE.

Header Type 2

PB ADDRESSES

```
  0  1                 10 11      15
|--|------------------|---------|
|//|       NW          |    2    |
|--------------------------------|
|             PBA                |
|--------------------------------|
|              .                 |
|              .                 |
|              .                 |
|--------------------------------|
|             PBA                |
----------------------------------
```

PBA - PB' ADDRESS OF PB ADDRESS TO BE CORRECTED

Header Type 3

OWN/DATA VARIABLES

```
  0  1                 10 11      15
|--|------------------|---------|
|//|        W          |    3    |
|--------------------------------|
| B|           PBA               |
|--------------------------------|
|               .                |
|               .                |
|               .                |
|--------------------------------|
| B|           PBA               |
----------------------------------
```

PBA - PB' ADDRESS OF OWN VARIABLE POINTER TO BE CORRECTED

Header Type 4

DSDB/OWN/DATA/VALUES

```
  0  1                10 11        15
|---|----------------|-----------|
|///|       NW        |     4     |
|---------------------------------|
|              LD                 |
|---------------------------------|
| B |          IN                 |
|---------------------------------|
|                                 |
|         INITIAL VALUES          |
|                                 |
-----------------------------------
```

LD - LOGICAL WORD DISPLACEMENT
     IN OWN ARRAY FOR INITIAL VALUES
B - BYTE BIT-SET IMPLIES THAT LD IS
     TYPE BYTE AND THAT THE FIRST
     WORD OF THE INITIAL VALUE BLOCK
     IS A COUNT OF THE NUMBER OF BYTES
     IN THE INITIAL VALUE BLOCK
IN - INTEGRATION NUMBER - NUMBER OF
     TIMES THE BLOCK OF INITIAL VALUE
     IS TO APPEAR IN THE SECONDARY BD -
     1->NO DUPLICATION,
     2->DUPLICATION, ETC

Header Type 5

PUST

```
  0  1                 10 11      15
|--|------------------|---------|
|//|        NW         |    5    |
|--------------------------------|
|             PBA                |
|--------------------------------|
|                                |
|        INITIAL VALUES          |
|                                |
----------------------------------
```

PBA - PB' ADDRESS OF LINKED LIST OF POINTERS TO BE INITIALIZED WITH DB ADDRESS OF PUST (SAME LIST FORMAT AS FOR FORMAT STRINGS) A PBA of -1 INDICATES NO FIX-UPS.

NOTE: ALL REFERENCES TO THE PUST INCLUDE THE FOUR-WORD HEADER THAT IS APPENDED BY THE SEGMENTER. THESE WORDS ARE NOT PRESENT IN THE HEADER; THEY ARE AUTOMATICALLY ALLOCATED AND INITIALIZED BY THE SEGMENTER.

Header Type 6

GLOBAL VARIABLES

```
 0 1           7 8    10 11   15
|--|------------|-------|-------|
|//|     NW            |   6   |
|-------------------------------|
|             TN                |
|-------------------------------|
|    DBA      |/////////|  NC   |
|-------------------------------|
|  CHAR.1     |    CHAR. 2      |
|-------------------------------|
|             .                 |
|             .                 |
|             .                 |
|-------------------------------|
|  CHAR. NC   |////////////////|
---------------------------------
```

Header Type 7

EXTERNAL VARIABLES

```
 0 1 2 3 4 5 6 7 8     10 11   15
|--|-|-|-|-|-|-|-|--------|-------|
|//|        NW           |   7   |
|---------------------------------|
|              TN                 |
|---------------------------------|
| M|//////|  NC   |    CHAR. 1    |
|---------------------------------|
|              .                  |
|              .                  |
|              .                  |
|---------------------------------|
|   CHAR. NC    |/////////////////|
|---------------------------------|
|              DA                 |
|---------------------------------|
|              PBA                |
|---------------------------------|
|              .                  |
|              .                  |
|              .                  |
|---------------------------------|
|              PBA                |
-----------------------------------
```

PBA-PB' address of linked lists of instructions to be repaired; lower 8 bits of inst. used as neg. displacement to next instruction; a link of 0 terminates the list.

M -Monitored variable bit;set if variable is being monitored by debug.

DA -Logical word disp. in PUST; lower 8 bits of word will be init. with prim.DB address of variable;DA is present if M=1.

NOTE:PBA of -1 implies null list

Header Type 8

PRIMARY DB

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
|-|-|-|-|-|-|-|-|-|-|--|--|--|--|--|--|
|/|           NW         |     8      |
|-------------------------------------|
|U | U  | U  | U  | U  | U  | U  | U  |
| 0|  1 |  2 |  3 |  4 |  5 |  6 |  7 |
|-------------------------------------|
|                 .                   |
|                 .                   |
|                 .                   |
|-------------------------------------|
|U   |U   |U   |U   |U   |////////////|
| N-5| N-4| N-3| N-2| N-1|////////////|
|-------------------------------------|
|                                     |
|                                     |
|          INITIAL VALUES             |
|                                     |
|                                     |
---------------------------------------
```

U - ADDRESS BITS
00 IF NO ADDRESS
01 IF NO ADDRESS
10 IF WORD ADDRESS IN SECONDARY DB
11 IF BYTE ADDRESS IN SECONDARY DB

N - NWPDB

NOTE: INITIAL ADDRESSES THAT ARE SECONDARY DB ADDRESSES ARE 0

RELATIVE (I.E., THEY ARE LOGICAL DISPLACEMENTS IN SECONDARY DB).

Header Type 9

COMMON VARIABLES

```
 0 1 2 3 4 5 6 7 8     10 11   15
|--|-|-|-|-|-|-|-|--------|-------|
|//|        NW            |   9   |
|---------------------------------|
|             NWC                 |
|---------------------------------|
|////////|  NC   |    CHAR. 1     |
|---------------------------------|
|              .                  |
|              .                  |
|              .                  |
|---------------------------------|
|   CHAR. NC    |/////////////////|
|---------------------------------|
| B| M|         NL                |
|---------------------------------|
|              LD                 |
|---------------------------------|
|              DA                 |
|---------------------------------|  -----
|              PBA                |    |
|---------------------------------|    |
|              .                  |    |
|              .                  |   NL
|              .                  |    |
|---------------------------------|    |
|              PBA                |    |
|---------------------------------|  -----
|                                 |
|              .                  |
|              .                  |
|              .                  |
|                                 |
|---------------------------------|
| B| M|         NL                |
|---------------------------------|
|              LD                 |
|---------------------------------|
|              DA                 |
|---------------------------------|
|              PBA                |
|---------------------------------|
|              .                  |
|              .                  |
|              .                  |
|---------------------------------|
|              PBA                |
-----------------------------------
```

Header Type 9 (cont.)

NWC - NUMBER OF WORDS IN COMMON ARRAY

NC - NUMBER OF CHARACTERS IN COMMON NAME- IF BLANK COMMON 4 COM'

DA - LOGICAL WORD DISP. IN PUST - LOWER 8 BITS OF WORD WILL BE INIT. WITH PRIM. DB ADDRESS OF VARIABLE - NOTE DA IS PRESENT IF M = 1

B - BYTE BIT
0 IF THE PRIMARY DB POINTER TO BE ALLOCATED AND INITIALIZED AND LD ARE OF TYPE WORD
1 IF TYPE BYTE

M - MONITORED VARIABLE BIT - SET IF VARIABLE IS BEING MONITORED BY DEBUG

NL - NUMBER OF ADDRESS LISTS FOR VARIABLE

LD - LOGICAL DISPLACEMENT OF VARIABLE IN COMMON ARRAY

PBA - PB' ADDRESS OF LINKED LISTS OF INSTRUCTIONS TO BE REPAIRED LOWER 8 BITS USED AS NEGATIVE DISPLACEMENT TO NEXT INSTRUCTION A LINK OF 0 TERMINATES THE LIST

PBA = -1 INDICATES NO FIX-UPS

Header Type 10

LOGICAL UNITS

```
 0                      10 11   15
|--|--------------------|-------|
|//|         8          |  10   |
|-------------------------------|
|                               |
|                               |
|                               |
|            BIT MAP            |
|                               |
|                               |
|                               |
---------------------------------
```

BIT MAP - BIT MAP OF LOGICAL UNITS
          REFERENCED; BIT 0
          CORRESPONDS TO LU 0, ETC.
          (1 LESS THAN OR EQUAL TO LU
 LESS THAN OR EQUAL TO 99)

Header Type 11

FORMAT STRING

```
 0                      10 11   15
|--|--------------------|-------|
|//|         NW         |  11   |
|-------------------------------|
|              PBA              |
|-------------------------------|
|              NC               |
|-------------------------------|
|   CHAR. 1     |   CHAR. 2     |
|-------------------------------|
|               .               |
|               .               |
|               .               |
|-------------------------------|
|   CHAR. NC    |///////////////|
---------------------------------
```

PBA - PB' ADDRESS OF LINKED LIST OF
      POINTERS TO BE INITIALIZED
      LOWER 14 BITS OF WORD USED
      AS NEGATIVE DISPLACEMENT TO
      NEXT POINTER - BIT 0 SET
      MEANS THAT THE POINTER IS TO
      BE TYPE BYTE - A LINK OF 0
      TERMINATES THE LIST.

RL File Format

```
    ------------                                             ------------
   0|   LID    |0  LOADER ID                               0|          |
    |----------|                                            |          |
   1|   FL     |1  FILE LENGTH (IN RECORDS)                 |  RECORD  |
    |----------|                                            |    0     |
   2|   NS     |2  NR. SECTIONS                             |          |
    |----------|                                             ------------
   3|          |3
    |----------|
   4|          |4
    |   SAXL   |   S.A. EXTERNAL SET LIST
   5|          |5
    |----------|                                             ------------
   6|          |6                                          1|          |
    |          |                                            |          |
   7|          |7                                           | FREE MAP |
    |          |                                            |    0     |
  10|          |8                                           |          |
    |          |                                             ------------
  11|          |9
    |          |
  12|          |10
    |          |
    |          |                                             ------------
    |          |                                         NS |          |
    |          |                                            |          |
    |          |                                            | FREE MAP |
    |          |                                            |   NS-1   |
    |          |                                            |          |
    |          |                                             ------------
    |          |  NOTE: UNINITIALIZED FIELDS ARE
    |          |        RESERVED FOR FUTURE USE AND
    |          |        SHOULD BE ZERO.
    |          |                                             ------------
    |          |                                        NS+1|          |
    |          |                                            |          |
    |          |                                            |          |
    |          |                                            |AVAILABLE |
    |          |                                            |          |
    |----------|                                            |          |
  41|   HL     |33 S.A. HASH LIST 0                         |          |
    |    0     |                                            |          |
    |----------|                                            |          |
    |    .     |                                            |          |
    |    .     |                                            |          |
    |    .     |                                            |          |
    |----------|                                            |          |
 177|   HL     |127 S.A. HASH LIST 94                       |          |
    |    94    |                                            |          |
    ------------                                             ------------
```

Storage Management

FILE SPACE IS MANAGED IN TERMS OF 32 WORDS BLOCKS (4 BLOCKS PER 128 WORD RECORD).

FREE SPACE (BLOCKS) IS ACCOUNTED FOR IN A BIT MAP, WHICH IS PARTITIONED INTO RECORDS (2K BLOCKS PER SECTION). A 0 INDICATES THAT A BLOCK IS USED, A 1 INDICATES THAT IT IS FREE.

FILE SPACE IS ALSO PARTITIONED INTO 512 RECORD SECTIONS (64 MAX. SECTIONS, 2K BLOCKS PER SECTION, 1 MAP PER SECTION). THE NUMBER OF SECTIONS IN A FILE IS NS=(FL+511) & LSR(9). THE FIRST NS RECORDS FOLLOWING RECORD 0 (RECORDS 1 TO NS) ARE RESERVED FOR THE SECTION MAPS.

A COMPLETE FILE ADDRESS WOULD HAVE THE FOLLOWING CONFIGURATION:

```
 0 1 2 3 4 5 6                15 16                26 27       31
|-----------|-------------------|-------------------|-----------|
|           |      SECTION      |       BLOCK       | DISPLCMT  |
-----------------------------------------------------------------
```

FILE (WORD) ADDRESS
DOUBLE WORD

Entry Point Directory

```
----------          ----------          ----------          ----------
|   HL   |------>|   LINK   |-->...-->|   LINK   |-->...-->|    0     |
----------          |--------|          |--------|          |--------|
                    |  USED  |          |  USED  |          |  USED  |
                    |--------|          |--------|          |--------|
                    |        |          |        |          |        |
                    |        |          |        |          |        |
                    |        |          |        |          |        |
                    |        |          |        |          |        |
                    |--------|          |--------|          |--------|
                    |////////|          |////////|          |////////|
                    |////////|          |////////|          |////////|
                    ----------          ----------          ----------
```

THE DIRECTORY IS PARTITIONED INTO 95 HASH LISTS (SAME HASH FUNCTION AS USL); EACH HASH LIST IS A LINKED LIST OF RECORDS.

EACH RECORD CONTAINS A SUCCESSOR LINK (RECORD #) AND A USED SPACE COUNT. A LINK OF 0 TERMINATES A LIST. WHEN A RECORD IS VOID OF ENTRIES (USED=2), ITS SPACE IS RETURNED TO THE FREE STORAGE AREA.

Typical Directory Entry

```
 0   1   2   3   4567   8                      15

|---|---|---|---|------|------------------------|
| S | U | I |///|  NC  |        CHAR. 1         |
|-----------------------------------------------|
|                      .                        |
|                      .                        |
|                      .                        |
|-----------------------------------------------|
|       CHAR. NC       |////////////////////////|
|-----------------------------------------------|
|                                               |
|               S.A. INFO BLOCK                 |
|                                               |
|-----------------------------------------------|
|                  S.A. ENTRY                   |
|-----------------------------------------------|
| F | W |            NW CODE                    |
|-----------------------------------------------|
|  LC   |      NP      |          CN            |
|-----------------------------------------------|
|                      TN                       |
|-----------------------------------------------|
|                   PARM. 1.                    |
|-----------------------------------------------|
|                      .                        |
|                      .                        |
|                      .                        |
|-----------------------------------------------|
|                   PARM. NP                    |
-------------------------------------------------
```

S - SECONDARY ENTRY POINT BIT - SET IF
    THE ENTRY POINT WAS ORIGINALLY A
    SECONDARY ENTRY POINT.

U - UNCALLABLE BIT - SET IF ENTRY POINT
    IS UNCALLABLE.

I - PRIVILEGED MODE BIT - SET IF CODE
    MODULE IS TO BE RUN IN PRIVILEGE MODE.

LC is (0:2)...Level of Checking
    0 = No checking
    1 >= Check for procedure type
    2 >= Check for # parameters
    3 >= Check for parameter type
NP is (2:6) is # parameters

Procedure Information Block

```
 0                             15
|-------------------------------|  ---------------
|            NW INFO            |
|-------------------------------|               |
|            NW CODE            |               |
|-------------------------------|               |
|        # ENTRY POINTS         |               |
|-------------------------------|  -------      |
|                               |               |
|                               |     |         |
|          CODE MODULE          |    NWC        |
|                               |     |         |
|                               |               |
|-------------------------------|  -------      |
|                               |               |
|           EXTN LINK           |               |
|                               |               |
|-------------------------------|               |
|             TPDB              |               |
|-------------------------------|               |
|             TSDB              |               |
|-------------------------------|               |
|             NWSDB             |              NWI
|-------------------------------|               |
|            HEADER             |               |
|-------------------------------|               |
|                               |               |
|            HEADER             |               |
|                               |               |
|-------------------------------|               |
|                               |               |
|               .               |               |
|                               |               |
|               .               |               |
|                               |               |
|-------------------------------|               |
|                               |               |
|            HEADER             |               |
|                               |               |
|-------------------------------|               |
|              -1               |
---------------------------------  ---------------
```

ALL HEADERS FOR THE PROCEDURE ARE APPENDED TO THE INFO BLOCK. THE HEADER SETS (EXTERNAL LISTS) ARE LINKED BY INCREASING FILE ADDRESS; A LINK OF %17777777777D TERMINATES THE LIST.

Headers

```
 0   1   2   3   4567   8        10 11        15
|---|---|---|---|------|----------|------------|
|///|              NW             |      1     |
|----------------------------------------------|
| F | W |        NW CODE                       |
|----------------------------------------------|
|                                              |
|            S.A. INFO BLOCK                   |
|                                              |
|----------------------------------------------|
|               S.A. ENTRY                     |
|----------------------------------------------|
|                  PBA                         |
|----------------------------------------------|
| S | U | I |///| NC  |       CHAR. 1          |
|----------------------------------------------|
|                     .                        |
|                     .                        |
|                     .                        |
|----------------------------------------------|
|       CHAR. NC      |////////////////////////|
|----------------------------------------------|
|   P   |     NP      |         CN             |
|----------------------------------------------|
|                     TN                       |
|----------------------------------------------|
|                  PARM. 1                     |
|----------------------------------------------|
|                     .                        |
|                     .                        |
|                     .                        |
|----------------------------------------------|
|                  PARM. NP                    |
------------------------------------------------
```

F - SET IF FATAL ERROR
W - SET IF NON-FATAL ERROR
S - SATISFIED BIT - SET IF EXTERNAL IS
    SATISFIED WITHIN RL.
U - UNCALLABLE BIT
I - PRIVILEGED BIT

ALL HEADERS ARE THE SAME AS IN A USL EXCEPT FOR THE PCAL HEADER.

CHAPTER 10 PREPARED OBJECT CODE

Program File Format

```
   ------------
 0|  FLAGS   | 0
  |----------|
 1|   NS     |1      NUMBER OF CODE SEGMENTS
  |----------|
 2|   GS     |2      GLOBAL SIZE (DB TO QI) IN WORDS
  |----------|
 3|   SAG    |3      GLOBAL AREA RECORD #
  |----------|
 4|   SAS    |       SEGMENT SET RECORD # (EACH SEG. STARTS IN NEW RECORD)
  |----------|
 5|   ISS    |5      INITIAL STACK SIZE IN WORDS
  |----------|
 6|   IDLS   |6      INITIAL DL SIZE IN WORDS
  |----------|
 7|   MAXD   |7      MAX. DATA SEGMENT SIZE (DL TO Z) IN WORDS
  |----------|
10|   SAE    |8      ENTRY POINT LIST RECORD #
  |----------|
11|   SSEG   |9      STARTING SEGMENT #
  |----------|
12|   SADR   |10     PRIN. ENTRY PT PB ADDRESS
  |----------|
13|  SASTLT  |11     DB ADR. OF STLT (-1 IF NO STLT)
  |----------|                (STLT=Segment Length Table)
14|  SAFLUT  |12     DB ADR. OF FLUT (-1 IF NO FLUT)
  |----------|
15|   SAX    |13     EXTERNAL LIST RECORD #
  |----------|
16|   SSTT   |14     PRIN. ENTRY PT SST #
  |----------|
17|  SATC    |15     STARTING ADDRESS OF TRAPCOM'
  |----------|
20|  SAPMAP  |16     STARTING RECORD OF PMAP INFO
  |----------|
21|  SASI    |17     STARTING RECORD OF SYMBOLIC ITEMS
  |----------|
22|  FLAGS2  |19
  |----------|
23|  CKSUM   |19     TOTAL CHECKSUM OF ALL SEGMENTS
  |----------|
24|          |20     NOTE : ALL UNUSED WORD ARE RESERVED FOR
  |----------|              FUTURE USE AND SHOULD BE SET TO
25|          |21            ZERO.
  |----------|
26|          |22
  |----------|
```

Program File Format (Cont.)

```
    |             |


    |-------------|
  27|             |23
    |-------------|
  30|             |24
    |-------------|
  31|             |25
    |-------------|
  32|             |26
    |-------------|
  33|             |27
    |-------------|
    | CST  | CST  |
  34|    o |    1 |28 \
    |-------------|   |
    |             |   |
    |-------------|   >   CST REMAPPING ARRAY
    | CST  |//////|   |
    |    n |//////|   |
    |-------------|   /
    |P|S|   SL    |   \
    | | |     o   |K  |
    |-------------|   |
    |      .      |   |
    |      .      |   >   SEGMENT DESCRIPTOR ARRAY
    |      .      |   |
    |-------------|   |
    |P|S|   SL    |   |
    | | |     n   |   |
    |-------------|   /
    |             |
    |             |
    |             |
    |             |
    |             |
    |             |
    |             |
    |             |
    |             |
    |             |
    |             |
   L|             | L
     --------------
```

P-PRIVILEGED MODE
S-Segment STT format: 0=> old format, 1=> new (extended) format
N=NS -1
K=28 + (NS +1) & LSR (1)
L=((28 + NS + (NS + 1)&LSR(1) + 127)/128)128 - 1

Flags

```
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
|-|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|
|F|W |Z |P |//|//|  |BA|IA|PM|   |   | MR|///| DS| PH
------------------------------------------------------
```

F - FATAL ERROR IN PROGRAM
W - NON-FATAL ERROR IN PROGRAM
Z - ZERO UNIT DL AREA
P - SET IF ANY SEG IS PRIVILEGED MODE (IF NOT SET NORMAL= NONPRIV MODE)

```
              CAPABILITIES

             /              BATCH ACCESS (9)   [BA]
             |
             |        INTERACTIVE ACCESS (8)  [IA]
             |
             |          PRIVILEGED MODE (7)  [PM]
             |
ACCESS TO    |
GENERAL     <
RESOURCES    |
             |
             |               MULTIPLE RINS (4)  [MR]
             |
             |
             |         EXTRA DATA SEGMENT (2)  [DS]
             |
             \           PROCESS HANDLING (1)  [PH]
```

Flags2

```
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
|-|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|
|T|K |            RESERVED                            |
------------------------------------------------------
```

T - PATCH AREA EXISTED IN ALL CODE SEGMENTS
K - CHECKSUM VALID

CST Remapping Array

CONTAINS THE LAST CST NUMBERS ASSIGNED TO THE SEGMENTS; INDEXED BY SEGMENT NUMBER. WHEN A PROGRAM FILE IS PREPARED, THE ARRAY IS INITIALIZED TO 0, 1...,N. THIS ARRAY IS USED TO RE-ESTABLISH INTRA-PROGRAM LINKAGE WHEN THE PROGRAM IS LOADED.

Segment Descriptor Array

CONTAINS THE SEGMENT LENGTH AND A FLAG INDICATING IF THE SEGMENT IS TO BE LOADED IN PRIV. MODE. INDEXED BY SEGMENT NUMBER. ALL SEGMENTS BEGIN ON A RECORD BOUNDARY. THE NUMBER OF RECORDS FOR A GIVEN SEGMENT IS (SL + 127) & LSR(7). THE RECORD NUMBER, SAS, OF SEGMENT N IS

```
SAS:=0
FOR I=0 TO N-1
 BEGIN
 SAS:=SAS + (SL(I) + 127)&LSR(7)
 END
```

Global Area Format

A SET OF RECORDS CONTAINING THE INITIAL VALUES FOR THE GLOBAL AREA OF THE DATA SEGMENT. THIS SET BEGINS AT RECORD SAG (WORD 3) AND CONSISTS OF (GS + 127) & LSR(7) RECORDS.

External List

```
0      7 8     15
|---|---|--------|                 ---     ---      ---
|///|NC | CHAR 1 | TYPICAL ENTRY     |       |        |
|----------------|                   |       |        |
|       .        |                   |       |        |
|       .        |               CHECK 0     |        |
|       .        |                   |       |        |
|----------------|                   |       |        |
|CHAR NC|////////|                   |  CHECK 1&2     |
|       NR       |                   |       |        |
|----------------|\                  |       |        |
| STT # | SEG #  ||                  |       |        |
|----------------||                  |       |        |
|       .        ||                  |       |        |
|       .        |>   NR             |       |        |
|       .        ||                  |       |    CHECK 3
|----------------||                  |       |        |
| STT # | SEG #  ||                  |       |        |
|-|--------------|/                  |       |        |
|LC| NP |   CN   |                   |       |        |
|-|--------------|                 ---     |        |
|       TN       |                           |        |
|----------------|                         ---        |
|     PARM 1     |                                    |
|----------------|                                    |
|       .        |                                    |
|       .        |                                    |
|       .        |                                    |
|----------------|                                    |
|                |                                    |
|----------------|                                    |
|    PARM NP     |                                    |
------------------                                  ---
        .
        .
        .
------------------
|       0        |   LIST TERMINATER
------------------
```

```
LC (0:2) = LEVEL OF CHECKING
           0 = NO CHECKING
           1 >= CHECK FOR PROCEDURE TYPE
           2 >= CHECK FOR # PARAMETERS
           3 >= CHECK FOR PARAMETER TYPE

NR         = NUMBER OF REFERENCES

NP (2:6)   = NUMBER OF PARAMETERS
```

Entry Point List

```
--------------------
|////| NC | CHAR 1 |
|------------------|
|        .         |
|        .         |
|        .         |
|------------------|
| CHAR NC |////////|
|------------------|
|     P.B. ADR     |
|------------------|
|      STT #       |
--------------------
         .
         .
         .
--------------------
|////| NC | CHAR 1 |
|------------------|
|        .         |
|        .         |
|        .         |
|------------------|
| CHAR NC |////////|
|------------------|
|     P.B. ADR     |
|------------------|
|      STT #       |
|------------------|
|        0         |  LIST TERMINATER
--------------------
```

NOTE THAT THE ENTRY POINT LIST MUST IMMEDIATELY FOLLOW THE EXTERNAL LIST.

Code Segment With Patch Area

```
--------
|      |
| CODE |
|      |
|------|
|      |
| PATCH|
| AREA |
|      |
|------|
|      |
| STT  |
|      |
--------
```

Patch Area

```
------------
| PROGRAM  |   4-WORD PROGRAM NAME
| NAME     |
|----------|
| SEGMENT  |   8-WORD SEGMENT NAME
| NAME     |
|----------|
|   //     |   1-WORD UNUSED
|----------|
| CHECKSUM |   1-WORD CHECKSUM
|----------|
| PREP TIME|   2-WORD PREP TIME
|----------|
|PATCH TIME|   2-WORD PATCH TIME
|----------|
|  PATCH   |
|  AREA    |
|----------|
| PALEN    |   1-WORD PATCH AREA LENGTH
|----------|
|          |
| STT      |
------------
```

PMAP Information

```
-----------------
|               |
|      PTT      |   PMAP TYPE TABLE
|               |
|---------------|
|               |
|      SPP      |   SEGMENT PMAP POINTERS
|               |
|---------------|
|               |
|      APD      |   ACTUAL PMAP DATUM
|               |
-----------------
```

PMAP Type Table

```
-----------------
|     PTTL      |   TYPE TABLE LENGTH
|---------------|
|     LPRO      |   LENGTH OF PMAP RECORD TYPE 0
|---------------|
|     LPR1      |   LENGTH OF PMAP RECORD TYPE 1
|---------------|
|       :       |
|       :       |
|---------------|
|     LPRn      |   LENGTH OF PMAP RECORD TYPE n
-----------------
```

NOTE : n = PTTL - 2

PMAP Records

Type 0 Segment PMAP Record

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
---------------------------------
|        0|  NC   |   char 1    |
|-------------------------------|
|               .               |
|               .               |
|               .               |
|-------------------------------|
| char NC       |///////////////|
|-------------------------------|
|   STT LEN     |   SEG NUM     |
|-------------------------------|
|      SEG LENGTH               |
---------------------------------
```

Type 1 Procedure PMAP Record

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
---------------------------------
|        1|  NC   |   char 1    |
|-------------------------------|
|               .               |
|               .               |
|               .               |
|-------------------------------|
| char NC       |///////////////|
|-------------------------------|
|H|/////////////////////////////|
|-------------------------------|
|        SA OF CODE             |
|-------------------------------|
|        CODE LENGTH            |
|-------------------------------|
|    PRIMARY ENTRY POINT ADDR   |
|-------------------------------|
|     COBOL TOOL BOX ID         |
|          LINK                 |
|-------------------------------|
|   TOOL BOX PROCEDURE ID       |
---------------------------------
```

Type 2 Secondary Entry PMAP Record

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
---------------------------------
|        2|  NC   |   char 1    |
|-------------------------------|
|               .               |
|               .               |
|               .               |
|-------------------------------|
| char NC       |///////////////|
|-------------------------------|
|H|/////////////////////////////|
|-------------------------------|
|  SECONDARY ENTRY POINT ADDR   |
|-------------------------------|
|    NUMBER OF ENTRY POINTS     |
---------------------------------
```

H : HIDDEN ENTRY FLAG

SL File Format

```
    --------
  0| LID  |0
   |------|
  1|  FL  |1 FILE LENGTH (IN RECORDS)
   |------|
  2|  EL  |2 EXTENT LENGTH (IN RECORDS)
   |------|
  3|      |3
   |------|
  4| NSEG |4 # SEGMENTS
   |------|
  5|      |5
   |------|
  6|      |6
   |------|
 7 | FRTL |7 S.A. OF FREE R.T. ENTRY LIST (-1 IF NONE)
   |------|
 10|      |8
   |------|
 11| NRT  |9 # REFERENCE TABLE ENTRIES
   |------|
 12|      |10
   |------|
 13|  NS  |11 # SECTIONS
   |------|
 14|      |12
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |      |
   |------|
 41| HLO  |33
   |------|
   |  .   |
   |  .   |
   |  .   |    NOTE:
   |------|    SHADED AND UNINITIALIZED FIELDS ARE
177|HL94  |127 RESERVED FOR FUTURE USE AND
   --------    SHOULD BE ZERO. HL = HASH LIST.
```

SL File Format (Cont.)

```
   0-----------
    |        |
    | RECORD |
    |   0    |
    |        |
    -----------

   1-----------
    |        |
    | RECORD |  <--- REFERENCE TABLE POINTERS
    |   1    |
    |        |
    -----------

   2-----------
    |        |
    |FREE MAP|
    |   0    |
    |        |
    -----------
        .
        .
        .
 NS+1-----------
    |        |
    |FREE MAP|
    |  NS-1  |
    |        |
    -----------

 NS+2------------
    |         |
    |         |
    |AVAILABLE|
    |         |
    |         |
    |         |
    |         |
    |         |
    |         |
    |         |
    |         |
    |         |
    -----------
```

Storage Management

FILE SPACE IS MANAGED IN TERMS OF 128 WORD BLOCKS (1 BLOCK PER 128 WORD RECORD).

FREE SPACE (BLOCKS) IS ACCOUNTED FOR IN A BIT MAP, WHICH IS PARTITIONED INTO RECORDS (2K BLOCKS PER SECTION). A 0 INDICATES THAT A BLOCK IS USED; A 1 INDICATES THAT IT IS FREE.

FILE SPACE IS ALSO PARTITIONED INTO 2048 RECORD SECTIONS (16 MAX. SECTIONS, 2K BLOCKS PER SECTION 1 MAP PER SECTION). THE NUMBER OF SECTIONS IN A FILE IS NS=(FL + 2047) & LSR(7). THE FIRST NS RECORDS FOLLOWING RECORDS 0, 1 (RECORDS 2 TO NS+1) ARE RESERVED FOR THE SECTION MAPS.

IF THE SECTION MAPS SPECIFY MORE SPACE THAN IS POTENTIALLY AVAILABLE, THOSE RECORDS BEYOND FLIMIT ARE MARKED AS "USED".

Entry Point Directory

```
----------      -----------       -----------       -----------
|  HL    |------>|  LINK   |->---->|  LINK   |->---->|    0    |
----------      |---------|       |---------|       |---------|
                |  USED   |       |  USED   |       |  USED   |
                |---------|       |---------|       |---------|
                |         |       |         |       |         |
                |         |       |         |       |         |
                |---------|       |---------|       |---------|
                |/////////|       |/////////|       |/////////|
                |/////////|       |/////////|       |/////////|
                -----------       -----------       -----------
```

THE DIRECTORY IS PARTITIONED INTO 95 HASH LISTS (SAME HASH FUNCTION AS USL); EACH HASH LIST IS A LINKED LIST OF RECORDS.

EACH RECORD CONTAINS A SUCCESSOR LINK (RECORD #) AND A USED SPACE COUNT. A LINK OF 0 TERMINATES A LIST. WHEN A RECORD IS VOID OF ENTRIES (USED=2), ITS SPACE IS RETURNED TO THE FREE STORAGE AREA.

THE HASH LIST HEAD POINTERS (HL IN THE DIAGRAM ABOVE) ARE IN RECORD 0 WORDS %41 TO %177.

Typical Directory Entry

```
 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
----------------------------------------------------------------
|///| U |///| P |      NC       |            CHAR 1            |
|--------------------------------------------------------------|
|                               .                              |
|                               .                              |
|                               .                              |
|--------------------------------------------------------------|
|       CHAR NC                 |//////////////////////////////|
|--------------------------------------------------------------|
|       STT #                   |            SEG #             |
|--------------------------------------------------------------|
| LC   |          NP            |            CN                |
|--------------------------------------------------------------|
|                              TN                              |
|--------------------------------------------------------------|
|                            PARM 1                            |
|--------------------------------------------------------------|
|                               .                              |
|                               .                              |
|                               .                              |
|--------------------------------------------------------------|
|                            PARM NP                           |
|--------------------------------------------------------------|
```

LC is (0:2)...Level of Checking
0 = No checking
1 >= Check for procedure type
2 >= Check for # parameters
3 >= Check for parameter type
NP is (2:6) is # parameters

P - 0= Not permanently allocated
1= Permanently allocated

U - Uncallable bit - set if entry point is uncallable.

Code Segment Linkage Structure

```
 ----------------
|                |
|                |
|  CODE SEGMENT  |
|                |
|                |
|----------------|
|                |
| STT MAP ARRAY  |
|                |
|----------------|
|                |
|                |
| EXTERNAL LIST  |
|                |
|                |
 ----------------
```

EACH CODE SEGMENT OCCUPIES AN INTEGRAL NUMBER OF RECORDS. THIS BLOCK OF INFORMATION CAN BE SUBDIVIDED INTO THREE TABLES: THE CODE SEGMENT PROPER, AN STT SEGMENT MAP ARRAY, AND AN EXTERNAL LIST.

STT MAP ARRAY

A 1 BYTE X 256 BYTE ARRAY. IT IS INDEXED BY STT NUMBER AND RETURNS (IF THE STT CORRESPONDS TO AN EXTERNAL OF THE SEGMENT) THE SEGMENT NUMBER OF THE EXTERNAL AND 255 OTHERWISE. THIS ARRAY IS USED WHENEVER THE SEGMENT IS LOADED AND IS UPDATED WHENEVER THE SL IS BOUND BY THE SEGMENTER.

EXTERNAL LIST

A SYMBOLIC LIST OF THE EXTERNALS OF THE SEGMENT. EACH ENTRY CONTAINS INFORMATION ABOUT THE EXTERNAL: PARAMETER CHECKING LEVEL AND PARAMETER MATCHING INFORMATION, AND THE SEGMENT NUMBER AND STT NUMBER IF THE EXTERNAL IS SATISFIED WITHIN THE SL.

Code Segment Structure (Cont.)

```
 0 1 2 3 4567 8          15

|-|-|-|-|----|------------|
|                         |
|                         |
|                         |
|                         |
|                         |
|                         |
|      CODE SEGMENT       |
|                         |
|                         |
|                         |
|                         |
|                         |
|-------------------------|
|                         |
|                         |
|      STT MAP ARRAY      |
|                         |
|-------------------------|
|S|/|/|/| NC |   CHAR. 1  | S - SATISFIED BIT - SET IF EXTERNAL
|-------------------------|     IS SATISFIED WITHIN SL
|            .            |
|            .            |
|            .            |
|-------------------------|
|  CHAR. NC  |////////////|
|-------------------------|
|  STT #     |   SEG. #   |
|-------------------------|
| P |  NP    |     CN     |
|-------------------------|
|           TN            |
|-------------------------|
|        PARM. 1          |
|-------------------------|
|                         |
|                         |
|-------------------------|
|        PARM. NP         |
|-------------------------|
|            .            |
|-------------------------|
|            0            |  EXTERNAL LIST TERMINATOR
 -------------------------
```

Reference Table Structure

FOR EACH SEGMENT THERE IS A REFERENCE TABLE ENTRY OF 32 WORDS. THE REFERENCE TABLE ENTRIES ARE PACKED FOUR TO A RECORD. THE RECORDS CONTAINING THE REFERENCE TABLE ENTRIES ARE LISTED IN RECORD 1. THE RECORD CONTAINING REFERENCE TABLE ENTRY N IS REC 1 (N.(0 : 14)); THE FIRST WORD OF THE ENTRY IS REFTAB (N.(14 : 2) & LSL (5)).

WHEN A SEGMENT IS DELETED, THE REFERENCE TABLE ENTRY CORRESPONDING TO THE SEGMENT IS RELEASED. THESE FREE ENTRIES ARE LINKED TOGETHER IN A LIST; THE SEGMENT # IS USED AS A LINK AND IS PLACED IN THE FIRST WORD OF THE ENTRY.

WHEN A SEGMENT IS ADDED IT IS ASSIGNED A SEGMENT NUMBER (0 LESS THAN/EQUAL TO N LESS THAN/EQUAL TO 254); THE NUMBER IS THAT OF THE FIRST FREE REFERENCE TABLE ENTRY, OR, IF NONE ARE FREE, THE NEXT AVAILABLE REFERENCE TABLE ENTRY (CAUSING SPACE ALLOCATION FOR THE ENTRY).

Reference Table (256 Maximum Entries)

```
                                               TYPICAL ENTRY

 DREC. 1      R.T. REC.        0 1 2 3 4 5 6 7 8 9            15  %
 ----------   -----------     |-|-|-|-|-|-|-|-|-|-|--------------|
 |  RL    |-->|    E    |-->|P|N|       SEGMENT LENGTH          | 0
 |    0   |   |    0    |   |-------------------------------|
 |--------|   |---------|   |   SEGMENT ADDRESS (REC. #)     | 1
 |        |   |    E    |   |-------------------------------|
 |    .   |   |     1   |   | # REC'S FOR SEG. & EXTN. LIST | 2
 |    .   |   |---------|   |-------------------------------|
 |    .   |   |    E    |   |F|S|/|/|A|C|X|/|/| # ENTRY PTS.  | 3
 |        |   |     2   |   |-------------------------------|
 |--------|   |---------|   |            SAPMAP              | 4
 |  RL    |   |    E    |   |-------------------------------|
 |    63  |   |     3   |   |            SASI                | 5
 ----------   -----------   |-------------------------------|
 (FILE REC1)  (1 SECTOR)    |T|K|                            | 6
                            |-------------------------------|
 SEG.NAME -16 BYTE ARRAY    |         SI LENGTH              | 7
           WITH NO CHARAC-  |-------------------------------|
           TER COUNT AND    |                                | 10
           TRAILING BLANKS  |                                |
           ADDED.           |                                |
                            |          SEGMENT NAME          |
 REF.MAP -256 BIT ARRAY     |                                |
          (INDEXED BY SEG#);|                                |
          BIT SET IF SEG IS |-------------------------------|
          REFERENCED DIRECT-|                                | 20
          LY OR INDIRECTLY. |                                |
                            |                                |
 F   SEGMENT DELETED        |                                |
 S   EXTERNAL SATISFIED     |                                |
 A   PERMANENTLY ALLOCATED  |                                |
 C   CORE RESIDENT SEGMENT  |                                |
 X   MPE SEGMENT            |                                |
 P   PRIV.INST. IN SEGMENT  |       REFERENCED SEGMENTS      |
 N   SLSEGFLAG              |            BIT MAP             |
 T   PATCH FLAG             |                                |
 K   CHECKSUM FLAG          |                                |
                            |                                |
 SLSEGFLAG:                 |                                |
        = 0 => SEG STT IS IN|                                |
               OLD FORMAT   |                                |
        = 1 => SEG STT IS IN|                                |
               NEW FORMAT --|                                |
               EXTENDED CSTS|                                |
                            |                                |
                            ----------------------------------
```

Code Segment With Patch Area

```
 --------
 |      |
 | CODE |
 |      |
 |------|
 |      |
 | PATCH|
 | AREA |
 |      |
 |------|
 |      |
 | STT  |
 |      |
 --------
```

Patch Area (Cont.)

```
 ------------
 | SEGMENT  |     8-WORD SEGMENT NAME
 | NAME     |
 |----------|
 |   //     |     1-WORD UNUSED
 |----------|
 | CHECKSUM |     1-WORD CHECKSUM
 |----------|
 | PREP TIME|     2-WORD PREP TIME
 |----------|
 |PATCH TIME|     2-WORD PATCH TIME
 |----------|
 | PATCH    |
 | AREA     |
 |----------|
 | PALEN    |     1-WORD PATCH AREA LENGTH
 |----------|
 |          |
 | STT      |
 ------------
```

PMAP Information

```
 ----------------
 |              |
 |     PTT      |     PMAP TYPE TABLE
 |              |
 |--------------|
 |              |
 |              |
 |              |
 |     APD      |     ACTUAL PMAP DATUM
 |              |
 ----------------
```

PMAP Type Table

```
 ----------------
 |    PTTL      |     TYPE TABLE LENGTH
 |--------------|
 |    LPR0      |     LENGTH OF PMAP RECORD TYPE 0
 |--------------|
 |    LPR1      |     LENGTH OF PMAP RECORD TYPE 1
 |--------------|
 |      :       |
 |      :       |
 |--------------|
 |    LPRn      |     LENGTH OF PMAP RECORD TYPE n
 ----------------

 NOTE : n = PTTL - 2
```

PMAP Records

Type 0 Segment PMAP Record

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
---------------------------------
|          0|  NC   |   char 1   |
|-------------------------------|
|                .               |
|                .               |
|                .               |
|-------------------------------|
|  char NC       |///////////////|
|-------------------------------|
|   STT LEN      |    SEG NUM    |
|-------------------------------|
|         SEG LENGTH             |
---------------------------------
```

Type 1 Procedure PMAP Record

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
---------------------------------
|          1|  NC   |   char 1   |
|-------------------------------|
|                .               |
|                .               |
|                .               |
|-------------------------------|
|  char NC       |///////////////|
|-------------------------------|
|H|//////////////////////////////|
|-------------------------------|
|         SA OF CODE             |
|-------------------------------|
|          CODE LENGTH           |
|-------------------------------|
|    PRIMARY ENTRY POINT ADDR    |
|-------------------------------|
|     COBOL TOOL BOX ID          |
|          LINK                  |
|-------------------------------|
|   TOOL BOX PROCEDURE ID        |
---------------------------------
```

Type 2 Secondary Entry PMAP Record

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
---------------------------------
|          2|  NC   |   char 1   |
|-------------------------------|
|                .               |
|                .               |
|                .               |
|-------------------------------|
|  char NC       |///////////////|
|-------------------------------|
|H|//////////////////////////////|
|-------------------------------|
|  SECONDARY ENTRY POINT ADDR    |
|-------------------------------|
|     NUMBER OF ENTRY POINTS     |
---------------------------------
```

H : HIDDEN ENTRY FLAG

CHAPTER 11 LOADER

MPE Loader

The loader is a system process which will do loads sequentially. If a process needs code to be loaded, it will get the load process' SIR, fill loader communication table, and then awake the loader. Upon completion, the loader will return its status through the loader communication table, and then activate the waiting process.

Loader Segment Table Overview

Loader Segment Table consists of two DST's. The main one is DST %22 (LST). The other DST (XLST) has its DST number stored in SYSGLOB %226.

LST Overview

```
          --------------------------
      DB |                          |
         |                          |
               PRIMARY DB AREA
         |                          |
         |                          |
         |--------------------------|
 DB+2 -->|                          |
            LOADER COMMUNICATION
         |         TABLE            |
         |--------------------------|
 DB+7 -->|                          |
              SBUF (128 words)
         |                          |
         |--------------------------|
 DB+O -->|                          |
         |                          |
                 DIRECTORY

         |                          |
       Z |                          |
          --------------------------
```

XLST Overiew

```
          --------------------------
      DB |                          |
         |                          |
               PRIMARY DB AREA
         |                          |
         |                          |
         |--------------------------|
 DB+7 -->|                          |
              SBUF (128 words)
         |                          |
         |--------------------------|
 DB+O -->|                          |
         |                          |
                 DIRECTORY

         |                          |
       Z |                          |
          --------------------------
```

The above DST's has exactly the same primary DB area so that directory entry handling procedures can be used on both DST'S. XLST is the LST extension and is used to store the extension entry only. When a extension entry is needed, it is copied into the LST to eliminate frequent EXCHANGEDB. Note that XLST is capable for any types of entries. It is used for extension entry only for now. Also, some of the primary DB's in the XLST are not used. They are there just for the consistency.

Loader Segment Table Primary DB

```
    -----------------             |-----------------|
  0|     @DIR       |          16|       SO        |
   |----------------|            |-----------------|
  1|   DIR LEN      |          17|       SP        |
   |----------------|            |-----------------|
  2|     @LCT       |          20|       SQ        |
   |----------------|            |-----------------|
  3|     ENTP       |          21|       SR        |
   |----------------|            |-----------------|
  4|     ENTP1      |          22|       SS        |
   |----------------|            |-----------------|
  5|     ENTP2      |          23|       ST        |
   |----------------|            |-----------------|
  6|     ENTP3      |          24|HDFWLINK(TYPE 0)|
   |----------------|            |-----------------|
  7|     @SBUF      |                    :
   |----------------|            |-----------------|
 10|      SI        |            |HDFWLINK(TYPE 8)|
   |----------------|            |-----------------|
 11|      SJ        |            |HDBKLINK(TYPE 0)|
   |----------------|            |-----------------|
 12|      SK        |                    :
   |----------------|            |-----------------|
 13|      SL        |            |HDBKLINK(TYPE 8)|
   |----------------|            |-----------------|
 14|      SM        |            |                 |
   |----------------|            |      LCT        |
 15|      SN        |            |       :         |
   |----------------|            |       :         |
```

ENTPn : POINTERS POINT TO THE CURRENT ACCESSED ENTRY.
SBUF : UTILITY BUFFER. USUALLY CONTAINS PROGRAM FILE RECORD
O INFORMATION.
SI ST : UTILITY DB RELATIVE VARIABLES.
HDFWLINKs : HEAD OF FORWARD LINK FOR EACH TYPE.
HDBKLINKs : HEAD OF BACKWARD LINK FOR EACH TYPE.

Directory Entries

```
| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   GARBAGE(0)
|               FORWARD LINK                     |
|------------------------------------------------|
|               BACKWARD LINK                    |
|------------------------------------------------|
|                LENGTH                          |
|------------------------------------------------|
|                       |          0             |
|------------------------------------------------|
|                      GARBAGE                   |
|------------------------------------------------|

| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   SL FILE(1)
|               FORWARD LINK                     |
|------------------------------------------------|
|               BACKWARD LINK                    |
|------------------------------------------------|
|                LENGTH                          |
|------------------------------------------------|
|                       |          1             |
|------------------------------------------------|
|             FILE DISC ADDRESS                  |
|                                                |
|------------------------------------------------|
|                FILE PV INFO                    |
|------------------------------------------------|
| # ALLOCATED SEG       | # SEGLIST ENTRIES      |
|------------------------------------------------|
|                                                |
            SEG ARRAY ( 16 WORDS )
|                                                |
|------------------------------------------------| \
|    LOG SEG NUMBER     |            |A |C |X |M | |
|------------------------------------------------| |  SEGLIST ARRAY
|                  REFERENCE COUNT               | >  3 WORD ENTRY
|------------------------------------------------| |  PER ALLOCATED
|                PHYSICAL CST NUMBER             | |  SL SEG
|------------------------------------------------| /
|                       :                        |
                        :
|                       :                        |
|------------------------------------------------|
|                                                |
|------------------------------------------------|
|                                                |
|------------------------------------------------|
|                                                |
|------------------------------------------------|
```

Directory Entries (Cont.)

```
| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|   PROGRAM
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   FILE (2)
|              FORWARD LINK                      |
|------------------------------------------------|
|              BACKWARD LINK                     |
|------------------------------------------------|
|               LENGTH                           |
|------------------------------------------------|
|P |A |     | LIB        |         2             |
|------------------------------------------------|
|            FILE DISC ADDRESS                   |
|                                                |
|------------------------------------------------|
|            CST BLOCK INDEX                     |
|------------------------------------------------|
|                 SEGMAP DST                     |
|------------------------------------------------|
|             # PROCESS SHARING                  |
|------------------------------------------------|
| # SEG IN PROGRAM FILE |    # SLINFO AREA       |
|------------------------------------------------|
|                 PV FILE INFO                   |
|------------------------------------------------|
|           TRACE EXTERNAL PLABEL                |
|------------------------------------------------| \
|            SL SEARCH SEQUENCE                  | |
|------------------------------------------------| |
|            SL FILE DISC ADDRESS                | | SL INFO AREA
|                                                | > 19 WORD PER
|------------------------------------------------| | EACH SL FILE
|                                                | |
|             LIB SEG ARRAY (16 WORDS)           | |
|                                                | /
|------------------------------------------------|
|                       :                        |
|                       :                        |
|                       :                        |
|                       :                        |
|                       :                        |
|------------------------------------------------| \
|                PSEGMAP SIZE                    | |
|------------------------------------------------| |
|   LIB LOG SEG         |    SL INFO INDEX       | |
|------------------------------------------------| | PSEGMAP
|   LIB LOG SEG         |    SL INFO INDEX       | > ARRAY
|------------------------------------------------| |
|                       :                        | |
|------------------------------------------------| |
|   LIB LOG SEG         |    SL INFO INDEX       | /
|------------------------------------------------|
```

Directory Entries (Cont.)

```
| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|   LOADING(3)
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|              FORWARD LINK                      |
|------------------------------------------------|
|              BACKWARD LINK                     |
|------------------------------------------------|
|               LENGTH                           |
|------------------------------------------------|
|P |                    |          3             |
|------------------------------------------------|
|           FILE DISC ADDRESS                    |
|                                                |
|------------------------------------------------|
```

```
| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   WAITER(4)
|              FORWARD LINK                      |
|------------------------------------------------|
|              BACKWARD LINK                     |
|------------------------------------------------|
|               LENGTH                           |
|------------------------------------------------|
|P |                    |          4             |
|------------------------------------------------|
|           FILE DISC ADDRESS                    |
|                                                |
|------------------------------------------------|
|                  WAITING PIN                   |
|------------------------------------------------|
|                    UNUSED                      |
|------------------------------------------------|
```

Directory Entries (Cont.)

```
| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|   LOADED(5)
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|              FORWARD LINK                      |
|------------------------------------------------|
|              BACKWARD LINK                     |
|------------------------------------------------|
|               LENGTH                           |
|------------------------------------------------|
|P |                    |          5             |
|------------------------------------------------|
|           FILE DISC ADDRESS                    |
|                                                |
|------------------------------------------------|
|               LOAD PROCESS STATUS              |
|                                                |
|------------------------------------------------|
```

```
| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|   SHARER(6)
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|              FORWARD LINK                      |
|------------------------------------------------|
|              BACKWARD LINK                     |
|------------------------------------------------|
|               LENGTH                           |
|------------------------------------------------|
|P |                    |          6             |
|------------------------------------------------|
|                      PIN                       |
|------------------------------------------------|
|           FILE DISC ADDRESS                    |
|                                                |
|------------------------------------------------|
```

Directory Entries (Cont.)

```
| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|   EXTENSION(7)
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|              FORWARD LINK                      |
|------------------------------------------------|
|              BACKWARD LINK                     |
|------------------------------------------------|
|               LENGTH                           |
|------------------------------------------------|
|           |  LIB    |          7               |
|------------------------------------------------|
|                      PIN                       |
|------------------------------------------------|
|                 EXTENSION ID                   |
|------------------------------------------------|
|LOADPROC COUNT(LOADPROC)/LOG SEG#(ALLOCATEPROC)|
|------------------------------------------------|
|                   PLABEL                       |
|------------------------------------------------|
| # CHAR IN NAME         |                       |
|------------------------|                       |
|                                                |
|              PROCEDURE NAME                    |
|                                                |
|------------------------------------------------|
|                        | # SL INFO AREA        |
|------------------------------------------------|
|                                                |
|                                                |
|    SL INFO AREA (19 WORDS PER SL INFO ENTRY)   |
|                                                |
|                                                |
|------------------------------------------------|
|                 MCSTREFSIZE                    |
|------------------------------------------------| \
|N |                    |     MCSTIDX(1)         | |
|------------------------------------------------| |
|                       :                        | > MCSTREF ARRAY
|                       :                        | |
|------------------------------------------------| |
|N |                    |     MCSTIDX(m)         | /
|------------------------------------------------|
```

Directory Entries (Cont.)

```
| 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   LOADPROC
|                FORWARD LINK                     |   MASTER(8)
|-------------------------------------------------|
|                BACKWARD LINK                    |
|-------------------------------------------------|
|                 LENGTH                          |
|-------------------------------------------------|
|                       |           8             |
|-------------------------------------------------|
|                      PIN                        |
|-------------------------------------------------|
|    # SLID ENTRIES     |    # ACTIVE LOADPROCS   |
|-------------------------------------------------|
|                                                 |
|           EXT IDX TABLE (16 WORDS)              |
|                                                 |
|-------------------------------------------------|
|                                                 |
|           MCST IDX TABLE (16 WORDS)             |
|                                                 |
|-------------------------------------------------|
|                    SLID(1)                      | \
|-------------------------------------------------| |
|                       :                         | | REFERENCED
|                       :                         | > SL ARRAY
|                       :                         | |
|-------------------------------------------------| |
|                    SLID(m)                      | /
|-------------------------------------------------|
|                       | #MCST LOGSEG SIZE       |
|-------------------------------------------------|
|                                                 |
|-------------------------------------------------| \
|     LOG SEG #         |    SLID INDEX(1)        | |
|-------------------------------------------------| |
|              REFERENCE COUNT                    | |
|-------------------------------------------------| |
|                       :                         | | MCST LOGSEG
|                       :                         | > ARRAY
|                       :                         | | 2 WORDS PER
|-------------------------------------------------| | ENTRY
|     LOG SEG #         |    SLID INDEX(n)        | |
|-------------------------------------------------| |
|              REFERENCE COUNT                    | |
|-------------------------------------------------| /
```

Loader Cache

SYGLOB extension area + %72 contains DST number of cache
BUCKETSIZE = %52

Cache Data Segment Format

```
                          ------------------------
                         0|                      |
                         1|    HIT COUNTER       |
                          ------------------------
                         2|                      |
                         3|    MISS COUNTER      |
                          ------------------------
                         4|      BUCKET 0        |
                          ------------------------
        4+ BUCKETSIZE     |      BUCKET1         |
                          ------------------------
                                     .
                                     .
                                     .
                          ------------------------
4+94* BUCKETSIZE          |                      |
                          |      BUCKET 94       |
4+95* BUCKETSIZE -1       |                      |
                          ------------------------
```

Bucket Format

```
        -----------------
     0 |   Length of     |
       |   SLDIR1 +1     |
        -----------------
     1 | SLDIR 1         | Most recently referenced system SL
       |                 | directory entry from this SL directory
       |-----------------| bucket
       |  LENGTH OF      |
       |  SLDIR2 + 1     |
        -----------------
       | SLDIR 2         | Second most recently referenced entry
        -----------------
       .                 .
       .                 .
        -----------------
       |  LENGTH OF      |
       | SLDIRN + 1      |
        -----------------
BUCKET | SLDIRN          | Nth most recently referenced entry;if
SIZE-1 |-----------------| not complete then indicates end of
                           bucket
```

All bucket words are initialized to BUCKETSIZE +1, indicating no entries.

Loader Communication Table (LCT)

Form Incoming to Loader (Load/Allocate Program)

```
   | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0 | CMD | LIB | M|LD| L|  |///////////////////////|
   |-----|-----|--|--|--|--|-----------------------|
 1 |                     PIN                        |
   |-----------------------|------------------------|
 2 |        LDEV           |                        |
   |-----------------------|                        |
 3 |                DISC ADDRESS                    | CMD=loader cmd
   |------------------------------------------------|     0=load prgm
 4 |                                                |     1=load proc
   |                                                |     2=alloc prog
 5 |                                                |     3=alloc proc
   |                                                | LIB=library
 6 |                                                |     search
   |                                                |     0=SYS
 7 |                  UNUSED                        |     1=PUB
   |                                                |     2=GROUP
 8 |                                                |
   |                                                |
 9 |                                                | M=NONPRIV MODE
   |                                                | LD=LOAD DOMAIN
10 |                                                | L=LOAD MAP REQ.
   |------------------------------------------------|
11 |   WRITER  PCB  INDEX                           |
   |------------------------------------------------|
12 |                      |BA|IA|PM|     |MR|  |DS|PH| USER CAPABILITY
   |----------------------|--|--|--|-----|--|--|--|--|
13 |                                                |
   |                                                |
14 |           GROUP                                |
   |                                                |
15 |                   NAME                         |
   |                                                |
16 |                                                |
   |------------------------------------------------|
17 |                                                |
   |                                                |
18 |           ACCOUNT                              |
   |                                                |
19 |                   NAME                         |
   |                                                |
20 |                                                |
   |------------------------------------------------|
21 |                 PV INFO                        |
   |------------------------------------------------|
```

LCT (Cont.)

Form Incoming to Loader (Load/Allocate Procedure)

```
   | 0| 1| 2| 3| 4| 5| 5| 7| 8| 9|10|11|12|13|14|15|
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0 | CMD | LIB | M|LD| L|  |///////////////////////|
   |-----|-----|--|--|--|--|-----------------------|
 1 |                     PIN                        |
   |-----------------------|------------------------|
 2 |                EXTENSION ID                    |
   |------------------------------------------------|
 3 |  # CHAR IN NAME       |                        | CMD=loader cmd
   |-----------------------|                        |     0=load prgm
 4 |                                                |     1=load proc
   |                                                |     2=alloc prog
 5 |                                                |     3=alloc proc
   |                                                | LIB=library
 6 |              PROCEDURE NAME                    |     search
   |                                                |     0=SYS
 7 |                                                |     1=PUB
   |                                                |     2=GROUP
 8 |                                                |
   |                                                |
 9 |                                                | M=NONPRIV MODE
   |                                                | LD=LOAD DOMAIN
10 |                                                | L=LOAD MAP REQ.
   |------------------------------------------------|
11 |   WRITER  PCB  INDEX                           |
   |------------------------------------------------|
12 |                      |BA|IA|PM|     |MR|  |DS|PH| USER CAPABILITY
   |----------------------|--|--|--|-----|--|--|--|--|
13 |                                                |
   |                                                |
14 |           GROUP                                |
   |                                                |
15 |                   NAME                         |
   |                                                |
16 |                                                |
   |------------------------------------------------|
17 |                                                |
   |                                                |
18 |           ACCOUNT                              |
   |                                                |
19 |                   NAME                         |
   |                                                |
20 |                                                |
   |------------------------------------------------|
21 |                 PV INFO                        |
   |------------------------------------------------|
```

LCT (Cont.)

Form Returned (No Error)

```
   |---------------------------------------------|
 0 |M |MF|  STARTING SEGMENT NUMBER               |
   |---------------------------------------------|
 1 |                      0                      |
   |---------------------------------------------|
 2 |   LOAD MAP FLAG                             |
   |---------------------------------------------|
 3 |                     LDEV                    |
   |---------------------|-----------------------|
 4 |          DISC                               |
 5 |                   ADDRESS                   |
   |---------------------------------------------|
 6 |              TRACE LABEL (IF TRACE)         |
   |---------------------------------------------|
```

Form Returned (Error Occurred)

```
    ---------------------------------------------
 0 |            FILE SYSTEM ERROR #              |
   |---------------------------------------------|
 1 |              LOADER ERROR #                 |
   |---------------------------------------------|
```

Logical Segment Transform Table (LSTT)

When a process references any user SL segments, these segments are assigned logical segment numbers if the new mapping ucode is running. The LSTT provides a map mapping these logical segments into their physical segment numbers and having true STT's for the mapped segments. The LSTT is created by LOADER during the load time. It occupies an DST and the DST number is stored in PCB(15). If no user SL segment is referenced, the LSTT will not be needed, hence it will not be created.

The new mapping microcode depends on the existence of the LSTT for getting the physical segment number for a mapped segment. So the LSTT has to be included in process' locality list if there is an LSTT. Dispatcher will then bring the LSTT in before the process can be run. Also the bank and address for the LSTT belonging to the current running process are stored in sysglob cells (%221 and %222) during the launch time by the dispatcher. These cells are used by microcode for fast accessing the LSTT.

Logical Segment Transform Table (LSTT) (Cont.)

```
            +--------------------------------+
            |   # of Logical Segments        |
            +--------------------------------+
            |      Length of LSTT            |
            +--------------------------------+  ---
            |       Physical Segment #       |
            +--------------------------------+  Logical seg 1
      ---   |   Pointer to STT list          |
       |    +--------------------------------+  ---
       |    |       Physical Segment #       |
       |    +--------------------------------+  Logical seg 2
       |    |   Pointer to STT list          |
       |    +--------------------------------+  ---
       |    |              .                 |   .
       |    |              .                 |   .
       |    |              .                 |   .
       |    +--------------------------------+  ---
       |    |       Physical Segment #       |
       |    +--------------------------------+  Logical seg n
    --+--   |   Pointer to STT list          |  (Max 255)
    |  |    +--------------------------------+  ---
    |  |    |M|  STT #      |    SEG #       |
    |  |    +--------------------------------+  STT's for logical
    |  |    |M|  STT #      |    SEG #       |  segment 1
    |  |    +--------------------------------+  (if needed)
    |  |    |              .                 |
    |  |    |              .                 |
    |  |    +--------------------------------+
    |  |    |M|  STT #      |    SEG #       |
    |  |    +--------------------------------+
    |  -->  |    Total STT's for this seg    |
    |       +--------------------------------+  ---
    |       |              .                 |   .
    |       |              .                 |   .
    |       |              .                 |   .
    |       +--------------------------------+  ---
    |       |M|  STT #      |    SEG #       |
    |       +--------------------------------+  STT's for logical
    |       |M|  STT #      |    SEG #       |  segment n
    |       +--------------------------------+  (if needed)
    |       |              .                 |
    |       |              .                 |
    |       +--------------------------------+
    |       |M|  STT #      |    SEG #       |
    |       +--------------------------------+
    ---->   |    Total STT's for this seg    |
            +--------------------------------+  ---
```

CHAPTER 12 PRIVATE VOLUMES / SERIAL DISC

Mounted Volume Table (MVT AB) DST =53 (%65)

```
                          1 1 1 1 1 1
    0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
    |------------------------------|  -----------------
   0|  entry size   :  max entries | 0                 |
    |------------------------------|                   |
   1|  # of mounted volume sets    | 1                 |
    |------------------------------|                   |
   2|   ldev        :    DIRBASE   | 2 master volume of|
    |----------------             -|   SYS VS is always|
   3|    of SYSTEM volume set      | 3 ldev = 1.       |
    |------------------------------|                   |
   4|              0               | 4                 |
    |------------------------------|                   |
   5|              0               | 5                 |
    |------------------------------|                   |
    |                              |                   |-- entry 0
    -                              -                   |  (MVTABX = 0)
                                                       |
    -                              -                   |
    |                              |                   |
    |------------------------------|                   |
  17|              0               |21                 |
    |------------------------------|                   |
  18|              0               |22                 |
    |------------------------------|                   |
  19|              0               |23                 |
    |------------------------------|                   |
  20|              0               |24                 |
    |------------------------------|  -----------------
```

MVTAB (Cont.)

```
    |------------------------------|  -----------------
   0|1 |cycl | Dircsize/32         | 0                 |
    |------------------------------|                   |
   1| hvol  | nvol  |       ucnt   | 1                 |
    |------------------------------|                   |
   2|    ldev       |    DIRBASE   | 2 master volume   |
    |----------------              |   of volume set   |
   3|       of volume set          | 3 is on this ldev |
    |------------------------------|                   |
   4|      generation number       | 4                 |
    |------------------------------| ----              |
   5|    ldev       :     VTABX    | 5 |               |
    |------------------------------|   |- vol entry 0  |-- entry 1
   6|    dbms       :     vcnt     | 6 |  (double)     |  (MVTABX = 1)
    |------------------------------| ----              |
    |                              |                   |
    -              .               -                   |
    -              .               -                   |
    |                              |                   |
    |------------------------------| ----              |
  19|    ldev       :     VTABX    |23 |               |
    |------------------------------|   |- vol entry 7  |
  20|///////////////:     vcnt     |24 |  (double)     |
    |------------------------------| ---- -------------|
    |                              |
    |                              |
    -                              -
    |                              |
    |                              |
    |------------------------------|  -----------------|
    |                              |                   |
    |                              |                   |
    -                              -                   |-- entry n-1
    |                              |                   |  (MVTABX = n-1)
    |                              |                   |
    |------------------------------|  -----------------
```

MVTAB (Cont.)

```
    |------------------------------|  -----------------
   0|1 |cycl |  Dircsize/32        | 0                 |
    |------------------------------|                   |
   1| hvol  | nvol  |       ucnt   | 1                 |
    |------------------------------|                   |
   2|    ldev       :    DIRBASE   | 2                 |
    |----------------              |                   |
   3|       of volume set          | 3                 |
    |------------------------------|                   |
   4|      generation number       | 4                 |
    |------------------------------| ----              |
   5|    ldev       :     VTABX    | 5 |               |
    |------------------------------|   |- vol entry 0  |-- entry n
   6|    dbms       :     vcnt     | 6 |  (double)     |  (MVTABX = n)
    |------------------------------| ----              |
    |                              |                   |
    -              .               -                   |
    -              .               -                   |
    |                              |                   |
    |------------------------------| ----              |
  19|    ldev       :     VTABX    |23 |               |
    |------------------------------|   |- vol entry 7  |
  20|///////////////:     vcnt     |24 |  (double)     |
    |------------------------------| ---- -------------|
```

cycl - cyclical volume index (local VTABX) for disc space allocation

hvol - highest (ordinal) volume index (volume index being the volume set's local VTABX) of a mounted member of the volume set(class).

nvol - # of volumes mounted for the volume set(class).

ucnt - # of users having mounted the volume set.

dbms - directory bit map size (sectors).

vcnt - # of users having mounted the volume.

Private Volume User Table (PVUSER) DST =54 (66)

```
                          1 1 1 1 1 1
    0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
    |------------------------------|  -----------------
   0|     table size (words)       | 0                 |
    |------------------------------|                   |
   1|        # of entries          | 1                 |
    |------------------------------|                   |
   2|bitmask of MVTABX's represented| 2                |-- table head
    |------------------------------|                   |  (5 words)
  $3| maximum table size ( words ) | 3                 |
    |------------------------------|                   |
   4|      available pointer       | 4                 |
    |------------------------------|  -----------------|
    |   op mask     :    MVTABX    |     |             |
    |------------------------------|     |             |
    |          max users           |     |             |
    |------------------------------|     |             |
    |           # pins             |     |- entry head |
    |------------------------------|     |  (5 words)  |
    |   current size of entry      |     |             |
    |------------------------------|     |             |
  $ |     PV flags           |OP   |     |             |
    |------------------------------|  ----             |
    |           vmask              |     |             |
    |------------------------------|     |             |
    |            pin               |     |             |
    |------------------------------|     |             |
    |      user bind count         |     |             |
    |------------------------------|     |             |
    |      user mount count        |     |             |
    |------------------------------|     |             |
    |     system bind count        |     |- user entry 1|
    |------------------------------|     |             |
    |     system mount count       |     |             |
    |------------------------------|     |             |
    |      bind names count        |     |             |
    |------------------------------|     |             |
    | DST # of bind names segment  |     |             |
    |------------------------------|  ----             |
    |           vmask              |     |             |
    |------------------------------|     |             |
    |            pin               |     |             |-- volume set
    |------------------------------|     |             |   entry 1
    |      user bind count         |     |             |   (MVTABX = j)
    |------------------------------|     |             |
    |      user mount count        |     |             |
    |------------------------------|     |             |
    |     system bind count        |     |- user entry 2|
    |------------------------------|     |             |
    |     system mount count       |     |             |
    |------------------------------|     |             |
```

PVUSER (Cont.)

```
|------------------------------|      |                |
|       bind names count       |      |                |
|------------------------------|      |                |
| DST # of bind names segment  |      |                |
|------------------------------| ----                  |
~                              ~                       |
~               :              ~                       |
|                              |                       |
|------------------------------| ----                  |
|            vmask             |      |                |
|------------------------------|      |                |
|             pin              |      |                |
|------------------------------|      |                |
|       user bind count        |      |                |
|------------------------------|      |                |
|       user mount count       |      |                |
|------------------------------|      |                |
|      system bind count       |      |- user entry n|
|------------------------------|      |                |
|      system mount count      |      |                |
|------------------------------|      |                |
|       bind names count       |      |                |
|------------------------------|      |                |
| DST # of bind names segment  |      |                |
|------------------------------| -----------------|
|                              |
~                              ~

~                              ~
|                              |
|------------------------------| -----------------|
|   op mask     :     MVTABX   |                  |
|------------------------------|                  |
|                              |                  |
~                              ~                  |-- volume set
                                                  |   entry n
~                              ~                  |   (MVTABX = k)
|                              |                  |
|------------------------------| -----------------|
|                              |
|               a              |
|               v              |
|               a              |
~               i              ~
                l
~               a              ~
|               b              |
|               l              |
|               e              |
|                              |
|------------------------------|
```

Bind Names Data Segment

(Created and managed via PVUSER Table)

```
                         1 1 1 1 1 1
  0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
   |------------------------------| -----------
  0|     max segment length       | 0         |
   |------------------------------|           |
  1|   current segment length     | 1         |
   |------------------------------|           |
  2|              0               | 2         |
   |------------------------------|           |
   |                              |           |
   ~                              ~           |-- entry 0
                                              |
   ~                              ~           |
   |                              |           |
   |------------------------------|           |
  8|              0               |10         |
   |------------------------------| ----------|
  0|         bind count           | 0         |
   |------------------------------|           |
  1|                              | 1         |
   |-         G R O U P          -|           |
  2|                              | 2         |
   |-          N A M E           -|           |
  3|                              | 3         |
   |-                            -|           |
  4|                              | 4         |-- entry 1
   |------------------------------|           |
  5|                              | 5         |
   |-       A C C O U N T        -|           |
  6|                              | 6         |
   |-          N A M E           -|           |
  7|                              | 7         |
   |-                            -|           |
  8|                              |10         |
   |------------------------------| -----------
   |                              |
   |                              |
   ~                              ~
```

Bind Names Data Segment (Cont.)

```
   ~                              ~
   |                              |
   |                              |
   |------------------------------| -----------
  0|         bind count           | 0         |
   |------------------------------|           |
  1|                              | 1         |
   |-         G R O U P          -|           |
  2|                              | 2         |
   |-          N A M E           -|           |
  3|                              | 3         |
   |-                            -|           |
  4|                              | 4         |-- entry n
   |------------------------------|           |
  5|                              | 5         |
   |-       A C C O U N T        -|           |
  6|                              | 6         |
   |-          N A M E           -|           |
  7|                              | 7         |
   |-                            -|           |
  8|                              |10         |
   |------------------------------| -----------
   |               a              |
   |               v              |
   |               a              |
   ~               i              ~
                   l
                   a
   ~               b              ~
   |               l              |
   |               e              |
   |                              |
   |------------------------------|
```

Serial Disc Tables and Data Structures

Data Record Format

The primary purpose of the Serial Disc Interface (SDISC) is to adapt the undefined length transfers characteristic of magnetic tape to the fixed-length environment of a disc or integrated cartridge tape(ICT). To accomplish this, data is buffered within SDISC. The buffer is an integral number of sectors (blocks for the ICT) long. Files always start on a sector boundary, but data records within files may start anywhere and straddle sector boundaries. A record in the buffer is structured as follows:

```
+---------+-----------------------------+---------+
| record  |                             | record  |
| length  |            data             | length  |
| (bytes) |                             | (bytes) |
+---------+-----------------------------+---------+
```

The record length is always a one-word positive byte count which includes only the data portion of the record, not the length words themselves. Records within a file might be stored on the disc as follows:

```
+----+---------------------------+---          -----
| RL |///////////////////////////|               ^
+----+------+----+----+----------+               |
|///////////| RL | RL |//////////|               |
+-----------+----+--+-+--+----+--|           Sector N-1
|///////////////////| RL | RL |//|               |
+-------------------+----+----+--|               |
|////////////////////////////////|               v
+---+----+----+------------------+---          -----
|///| RL | RL |//////////////////|               ^
+---+----+--+-+--+----+----------+               |
|///////////| RL | RL |//////////|               |
+-----------+----+----++----+----+           Sector N
|/////////////////////| RL | RL |                |
+---------------------+----+-----+               |
|////////////////////////////////|               v
+-----+----+----+----------------+---          -----
|/////| RL | RL | .....          |
+-----+----+----+----------------+
```

The reason for the trailing byte count is to implement an easy way to backspace records.

End of File Format

Since files always start on a sector boundary, it follows that they also end on one. End of files consist of a 0 record length and 0-fill to the end the current sector as follows:

```
+---------------------------------+
|//////////////////////// RL RL /|
|/////////////////////////////////|
|//////// RL RL ///////////////// |          Sector N
|                     +-----------+
|////////////////| RL | 0          |
+---------------------+           |
|                                 |
|           Zero fill             |
+---------------------------------+---
```

In addition, an End-of-File entry is made in the Gap Table, so that files may be skipped by scanning Gap Table entries instead of serially scanning the data area. The Gap Table is described a few pages from now.

Contiguous Block Format

A serial disc, if it can do everything a magnetic tape can do, must also be a cold-load device. This means that machine microcode must be able to read a bootstrap channel program and the resident segments of INITIAL from the disc into memory. The microcode and channel programs cannot deal with the record length words which surround standard data records, so for them we have a structure, called a CONTIGUOUS BLOCK, which has the data without the length words. Information as to the length of each contiguous block must therefore be kept elsewhere, so there are Gap Table entries which hold the beginning and ending sector addresses of each contiguous block. This implies that each block must begin and end on a sector boundary. In this way they are similar to data files. To set contiguous blocks off from normal data, and to reach a sector boundary, a record length and fill character = %177777 is used, as follows:

```
+-------------------------------+---      -----
|/////// Previous records //////|            ^
|///////////////////////////////|            |
|            +------------------+            |
|////////| RL | -1              |         Sector N-1
+-------------+                 |            |
|                               |            |
|           -1 fill             |            |
|                               |            v
+-------------------------------+---      -----
|                               |            ^
|        Contiguous block       |         Sector N
|                               |            v
|                               +---      -----
|                               |            ^
|               +---------------+            |
|               |               |         Sector N+1
+---------------+               |            |
|           -1 fill             |            v
+-------------------------------+---      -----
```

Hole Format

Holes on the serial disc have the same format as contiguous blocks (that is, they start and end on sector boundaries with -1 fill characters as required). Starting with MPE version G.00.00, holes are obsolete and SDISC will not generate them. However, code has been left in SDISC to process any holes found on serial discs written with earlier versions of SDISC. Further details may be found in the Serial Disc IMS.

Gap Table Format

The Gap Table is a four-word header followed by a series of two-word device address entries. A permanent copy lives on the device, starting in sector 4, while a working copy lives in main memory. The copy in memory is posted to the disc only when a backspace or rewind operation occurs after writing (in other words, when the copy in main memory has changed). The length of the Gap Table is device-dependent according to the table below:

Device	Number of sectors (or ICT blocks)
HP7920	44
HP7925	106
HP7933/35	219
HP7902/9895	26
ICT	4 blocks ("S" cartridge) or 15 blocks ("L" cartridge)

The Gap Table looks like this:

```
   +----------------------------+
 0 | sector addr of load point |\
 1 |          unused            | \
 2 |          unused            | /-Gap Table header
 3 |          unused            |/
   +------+---------------------+
 4 | type |                     |
   +------+   Sector address    |   Entry (two words)
 5 |                            |
   +------+---------------------+
 6 | type |                     |
   +------+   Sector address    |   Entry (two words)
 7 |                            |
   +----------------------------+
                  .
                  .
                  .
```

The type field is bits 0, 1 and 2 of the first word. The eight possible types are:

0. End of File. The associated sector address contains one or more end of file fill characters (0) to fill out that sector. In the worst case (the previous record ended exactly at the end of the previous sector), the end of file sector contains all zeros.
1. End of data. The associated sector address is the last address of valid data plus 1, in other words, the next available address. In practice, such an entry is usually preceded by an end-of-file entry, since the EOD entry is written when you stop writing, and the file system will not let you backspace or rewind after writing without sending a Write End of File. An EOD entry is also written at the beginning of the Gap Table when new (unwritten) media is inserted. This prevents erroneous reading of blank media.

2. Beginning of Hole. The starting address of a "defective" area of the disc. Usually on a track boundary, but may be in mid-track if a contiguous block was being written when the "defect" was encountered. Obsolete, starting with MPE version G.00.00.
3. End of Hole. The corresponding ending address of the "defective" area. Always at a track boundary. Obsolete, starting with MPE version G.00.00.
4. Beginning of (contiguous) Block. The starting address of a contiguous block, exclusive of the -1 fill characters which may have been required to get us to a sector boundary. Unlike the End of File fill characters, there need not be any -1 characters if the previous record or contiguous block (with or without the trailing length word) ended exactly on a sector boundary.
5. End of (contiguous) Block. The address of the last sector containing contiguous block data. The sector may also contain -1 fill characters to get us to a sector boundary, but as with the beginning of block they are not required if the contiguous block ends exactly on a sector boundary.
6. End of Tape mark. The sector address of the simulated End of Tape reflector. This type is now written only to floppy discs for use by INITIAL's serial disc interface. When read by MPE's SDISC, it will be skipped no matter what device it is found on. This ensures compatibility with older serial discs.
7. End of Gap Table. No associated sector address. If you hit this while scanning the Gap Table, you've gone too far. In practice, this type is created whenever the Gap Table is cleared, by the simple device of initializing the table to -1.

SDISC Extra Data Segments

With insignificant exceptions, SDISC operates entirely in split-stack mode, that is, using an extra data segment for its working storage. Starting with MPE version G.00.00, there are two additional data segments used as no-wait data buffers. For the most part, our discussion here is restricted to the original data segment, now used only for variables, the Gap Table, and data buffer management.

The working storage extra data segment (XDS) is usually acquired by the external procedure ALLOCATE when the serial disc device is first assigned to a user as part of an FOPEN. The external procedure DEALLOCATE makes the XDS go away as part of its processing of the final FCLOSE against the device. The system program PVPROC may also acquire and release an XDS so that the tape label routines in LABSEG may also use SDISC for their work when DEVREC processes a device on-line interrupt. SDISC allocates the two data buffer segments as they are needed, then deallocates them as part of the Device Close processing.

In addition to the Gap Table already described, the XDS contains SDISC's global storage area, including the data buffer management areas (BUFFER'INFO), and a small buffer (called WORKTABLE). WORKTABLE holds the contents of the Serial Disc label sector when SDISC reads it in as part of its self-configuration. It also hold the Defective Tracks Table (MAC family discs) or Defective Sector Table (CS80 discs) while reassigning suspect or deleted tracks.

The three arrays in the XDS (WORKTABLE, BUFFER'INFO and GPT (Gap Table)) are all dynamically configured by SDISC as vanilla indirect arrays, such as might have been constructed by SPL. This is done by declaring the array names as pointers, then inserting appropriately computed element-0 addresses in them.

The extra data segment is organized as follows:

```
  +-------------------+   These twelve words are reserved
0 | WORDSPERSECTR     |   for use by ALLOCATE when the data
  |. . . . . . . . . .|   segment is created.  However, AL-
1 | SECTORSPERTRAK    |   LOCATE only stuffs the last five
  |. . . . . . . . . .|   of them.  We fill the first seven
2 | STARTADDRESS (BOT)|   ourselves with information we get
  |. . . . . . . . . .|   from the label sector.
3 | EOTSECTR (disc    |
  | address of simu-  |
4 | lated end of tape)|
  |. . . . . . . . . .|
5 | EODSECTR (last    |
  | sector of disc)   |   Simulates tape runoff.
6 |                   |
  |. . . . . . . . . .|
7 | JUSTALLOCATED     |   Tells us to initialize SDISC
  |. . . . . . . . . .|     parameters to BOT if true.
8 | WRITE RING        |   Simulation of tape write ring.
  |. . . . . . . . . .|
9 | FATALERROR        |   Disables SDISC when true.
  |. . . . . . . . . .|
```

```
10 | No longer used.   |
   |. . . . . . . . . .|
11 | MAX'DSEG'SIZE     |   Max size of our XDS, so we can
   +-------------------+     check that it's big enough.
   | SDISC global vari-|
   |   ables, including|
   |   array pointers. |
   +-------------------+
   | W                 |
   |   O               |   Length is 512 words.
   |     R             |
   |       K           |
   |         T         |
   |           A       |
   |             B     |
   |               L   |
   |                 E |
   +-------------------+
   | B         I       |
   |   U         N     |   Length is calculated as
   |     F         F   |     MAX'NUM'BUFFERS (currently 2) *
   |       F         O |     INFO'ENTRY'SIZE (currently 8).
   |         E         |
   |           R       |
   |             '     |
   +-------------------+
   | G                 |   Length varies with device, and is
   |   A               |     calculated by SDISC as part of its
   |     P             |     self-configuration.
   |                   |
   |         T         |
   |           A       |
   |             B     |
   |               L   |
   |                 E |
   +-------------------+
```

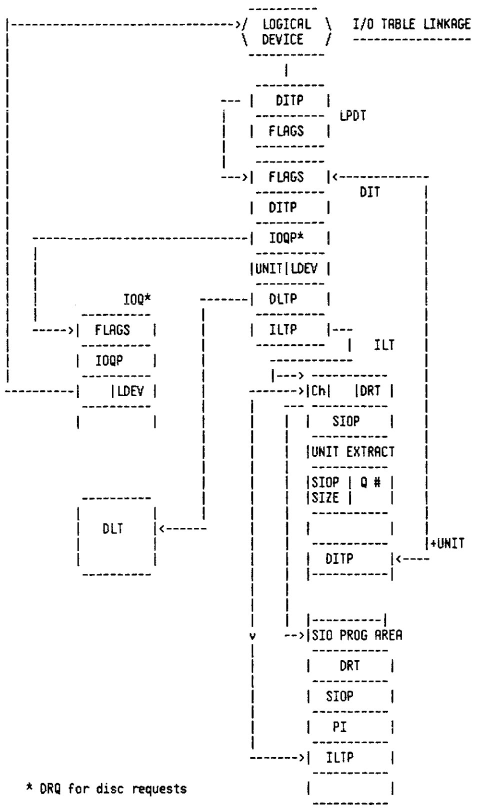
Serial Disc Organization

The disc is organized as follows:

```
+-------------------+
| Label sector      |   0   See expanded view in Chapter 3.
+-------------------+
| DTT/DSCT          |   1   DTT (MAC family) or DSCT (CS80).
+-------------------+
| Cold load         |   2   HP-IB cold load channel prog.
+-------------------+
| Soft dump         |   3   SOFTDUMP channel program.
+-------------------+
| Gap Table         |   4 to STARTADDRESS - 1.
|     .             |
|     .             |
+-------------------+
| Data              |   STARTADDRESS
|     .             |       .
|     .             |       .
|     .             |       to
|. . . . . . . . . .|       .
|     .             |   EOTSECTR
|. . . . . . . . . .|       .
|     .             |       to
|. . . . . . . . . .|       .
| Last data sector  |   EODSECTR
+-------------------+
```

CHAPTER 13 I/O

I/O Table Linkage



* DRQ for disc requests

Device Reference Table (DRT)

```
            HP-IB Systems
 ABS  -----------------------------
   8  |       Bank of DRT         |
      |---------------------------|     >----|
   9  |  Offset of DRT in Bank    |          |
      -----------------------------          |
                                             |
                                             |
        DRT ENTRY ON /33, /44                |
      -----------------------------          |
      |        SIOP               |     <----|
      |---------------------------|
      |        DBI                |
      |---------------------------|
      |        PI                 |
      |---------------------------|
      |        Channel Flags      |
      -----------------------------
```

SIOP - absolute address of SIO program
PI - interrupt handler plabel
DBI - this is the absolute address of the ILT

Driver Linkage Table (DLT)

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 |                       |DF|MC|CR|     |O |MTYP |
0|    QUEUE NUMBER       |  (SEE BELOW)         |    DPROC
 |----------------------------------------------|
1|              MONITOR PLABEL                  |    DMNTR
 |----------------------------------------------|
2|             INITIATOR PLABEL                 |    DINIT
 |----------------------------------------------|
3|             COMPLETOR PLABEL                 |    DCOMP
 |----------------------------------------------|
4|             INTERRUPT PLABEL                 |    DINTP
 |----------------------------------------------|
5|      DIT SIZE         |      DEVICE TYPE     |    DTYPE
 |----------------------------------------------|
6|          CS DRIVER EDITOR PLABEL             |
 |----------------------------------------------|
7|            INITIALIZATION PLABEL             |
 |----------------------------------------------|
```

There is one DLT for each type of driver. A pointer in the DIT allows different devices on a controller to have different drivers and interrupt handlers.

DPROC.QNUMB - This field contains the I/O process request queue number for type 2 drivers. Zero for all other types.
.(8:1).DRVRFRZN - Driver code frozen. Set by MAM when then the driver
(DF) code segment has been made present and frozen from a request from SIODM.
.(9:1).MAMERRORC- MAM Error on Code Makepresent
(MC)
.(10:1).CORERES - If set both initiator and completor code are core
(CR) resident.
.(14:2).DRVRTYPE- DRIVER/MONITOR TYPE
(MTVP)
0 - not used
1 - driver can be executed on any stack
2 - driver can be executed in the user process or in the I/O process identified by IDNUMB
3 - run only in process whose PCB number is in IDNUMB

DMNTR - I/O Monitor Plabel.
DINIT - Driver Initiator Procedure Plabel.

DCOMP - Driver Completor Procedure Plabel.
DINTP - Special interrupt handler Plabel. This procedure is called by GIP if ISPEC is set DFLAG. No other action is taken by GIP except to set the Interrupt Status in DSTAT.
DTYPE.DITSIZE - The length of the DIT in words for this driver.

Logical-To-Physical Device Table (LPDT)

DST = 13 (= %15)
SIR = 9 (= %11)

The LPDT has several fields which describe the state of a device. Some of these fields have the same meaning for all devices. Others are device dependent. All are described below.

There are two types of devices represented in the LPDT: real devices and virtual devices. A real device is one which has been configured into the system and is capable of performing input and/or output. A virtual device simulates some of the properties of a real device (for example a spooled line printer or an INP), but there is no physical I/O involved. The two main uses for virtual devices are for OPEN spooled devicefiles and certain communication devices (such as INP's).
A given virtual device entry is in use only while the devicefile it represents is open. When the file is FCLOSEd, the entry becomes available for another virtual device. This is the reason for the SYSDUMP/INITIAL configurator question MAX # OF OPEN SPOOLFILES--it needs to know how many virtual device entries to allocate to the LPDT (and to the LDT).
Entries in the LPDT are ordered by logical device number. The first word address of a real device entry is obtained by multiplying the LDN by the entry size. Except for the 0th entry, entries for which no logical device is configured on a given system are used for virtual device entries. Any remaining virtual device entries follow the last real device entry.

Entry 0

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
 0|          Number of entries in table           |
  +-----------------------------------------------+
 1|                 Entry size = 4                |
  +-----------------------------------------------+
 2|         DEVREC service request count          |
  +-----------------------------------------------+
 3|///////////////////////////////////////////////|
  +-----------------------------------------------+
```

Discussion:
Word 2 is incremented by a device driver whenever it sets the Device Ownership State field (below) to 2 (Service Requested). DEVREC decrements the count for each interrupt it services until the count reaches 0, at which time DEVREC hibernates.

-- CAUTION --

Device drivers must lock this table by DISABLE/ENABLEing, -NOT- by trying to acquire the LPDT SIR.

Typical Entry (Virtual Devices)

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
 0| 1|          Pointer to XDD subentry           |
  +--+--+--+--+--+--+--+--------+--+--+-----------+
 1|     |  |  |  |  |  |        |  |  |           |
  +-----+--+--+--+--+--+--------+--+--+-----------+
 2|///////////////////////////////////////////////|
  +--+--------------------------------------------+
 3|IO|////////////////////////////////////////////|
  +--+--------------------------------------------+
```

IO -- 0 for input, 1 for output.

Word 0, bit 0 is 1 for a virtual device, 0 for a real device. The fields in word 1 are the same, as applicable, as for the real device represented by a given virtual device. See below.

Typical Entry (All Real Devices)

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
 0| 0|////////////////////////////////////////////|
  +--+--+--+--+--+--+--+--------+--+--+-----------+
  |Devc | J|Da|  | D| I| End of |  |  |Au: Device |
 1|Owned| o|ta|  | u| n|  File  |  |  |to: Subtype|
  |State| b|  |  | p| t|Cndition|  |  |  :        |
  +-----+--+--+--+--+--+--------+--+--+-----------+
 2|        SYSDB-relative pointer to the DIT      |
  +-----------------------------------------------+
 3|///////////////////////////////////////////////|
  +-----------------------------------------------+
```

Discussion:

Word 1.(0:2) -- Device Ownership State:

0 -- Not owned by any process.

1 -- Owned by a process.

2 -- Service requested. Set by driver for unexpected interrupt, then wakes DEVREC.

3 -- Service granted. Set by DEVREC. Logon sequence is 0-2-3-1.

3 -- Device reserved (alternate use). Set during STARTSPOOL, spooler process sets to 1 when it gets started.

Word 1.(2:1) -- Device is Job/Session Accepting if true.

Word 1.(3:1) -- Device is Data Accepting if true.

Word 1.(5:1) -- Device is Duplicative if true (all devices except discs).

Word 1.(6:1) -- Device is Interactive if true (all devices except discs).

Word 1.(7:3) -- End of File condition:

0 -- No EOF detected.

1 -- Hardware EOF (e.g., tape mark).

2 -- :DATA record read.

3 -- :EOD record read.

4 -- :HELLO record read.

5 -- :BYE record read.

6 -- :JOB record read.

7 -- :EOJ record read.

Word 1.(12:4) -- Device subtype. See discussion for tape entry (below) for a description of the Auto bit (12:1).

The remaining bits in Word 1 are device-dependent and are described with their corresponding entry diagram.

Entry for Terminal-Like Devices

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
 0| 0|////////////////////////////////////////////|
  +--+--+--+--+--+--+--+--------+--+--+-----------+
  |Devc | J|Da|Ct| D| I| End of | B| L|    Device |
 1|Owned| o|ta|lY| u| n|  File  | r| o|    Subtype|
  |State| b|  |  | p| t|Cndition| k| g|           |
  +-----+--+--+--+--+--+--------+--+--+-----------+
 2|        SYSDB-relative pointer to the DIT      |
  +-----------------------------------------------+
 3|///////////////////////////////////////////////|
  +-----------------------------------------------+
```

Discussion (unique fields only):

Word 1.(4:1) -- CONTROL-Y is allowed and has been detected.

Word 1.(10:1) -- BREAK has been detected -OR- ignore BREAK if the C.I. is running.

Word 1.(11:1) -- The terminal is logging on. This bit is set by PROGEN and DEVREC when the logon sequence starts. If the bit is off when polled by INITJSMP, the terminal has disconnected. For now, only IOTERMO and HIOTERM support the use of this bit. Multipoint and DS pseudo-terminals do not.

Entry for Tape Drives

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
 0| 0|////////////////////////////////////////////|
  +--+--+--+--+--+--+--+--------+--+--+-----------+
  |Devc | J|Da| B| D| I| End of |  | A|Au: Device |
 1|Owned| o|ta| O| u| n|  File  |  | V|to: Subtype|
  |State| b|  | T| p| t|Cndition|  | R|  :        |
  +-----+--+--+--+--+--+--------+--+--+-----------+
 2|        SYSDB-relative pointer to the DIT      |
  +-----------------------------------------------+
 3|///////////////////////////////////////////////|
  +-----------------------------------------------+
```

Discussion (unique fields only):

Word 1.(4:1) -- BOT. Tape is at Load Point -OR- no tape mounted. Recording density may only be switched when this bit is true (for multiple density tape drives).

Word 1.(11:1) -- If true, DEVREC is performing Automatic Volume Recognition (AVR) on a tape (or PVPROC is doing the same on a serial disc), -OR- AVR is to be suppressed on job or data accepting devices.

Word 1.(12:1) -- Part of Device Subtype field. If true, device is allocated automatically when opened. If false, operator must allocate.

Entry for Disc Drives

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
 0| 0|////////////////////////////////////////////|
  +--+--+--+--+--+--+--+--------+--+--+-----------+
  |Devc | J|Da| N|Mt|RV| End of | S| F|    Device |
 1|Owned| o|ta| S|d |  |  File  |or| o|    Subtype|
  |State| b|  | D|PV|  |Cndition| F| r|           |
  +-----+--+--+--+--+--+--------+--+--+-----------+
 2|        SYSDB-relative pointer to the DIT      |
  +--+--+-----------------------------------------+
 3|//|SD|/////////////////////////////////////////|
  +--+--+-----------------------------------------+
```

Discussion (unique fields only):

Word 1.(0:2) -- Device Ownership State. May not be 1 (owned) for shared device (system volume or private volume). Serial and foreign discs are non-sharable and may be owned. See the full discussion of this field under Typical Entry, above.

Word 1.(4:1) -- If true, the disc is a nonsystem domain (private volume, serial disc or foreign disc) disc drive.

Word 1.(5:1) -- If true, disc is a mounted private volume.

Word 1.(6:1) -- If true, the disc is a reserved volume used to satisfy the requirements of a multiple volume private volume set.

Word 1.(10:1) -- If true, the disc is a physically and logically mounted serial or foreign disc. Bits 5 and 6 must be false.

Word 1.(11:1) -- If bit 10 is true, then 1 ==> foreign disc, 0 ==> serial disc.

Word 3.(1:1) -- If true, the device is currently being used as a serial disc (that is, it is allocated to a user as a serial disc). This bit duplicates a bit in the LDTX entry so that this information can be found in a system (memory-resident) table.

Logical Device Table (LDT)

Overview of Data Segment

```
DST 14 (= %16)   +----------------------------+<-----DST %16
SIR 10 (= %12)   |                            |
                 |   Logical Device Table     |
                 |                            |
                 |          (LDT)             |
                 |                            |
                 +----------------------------+
                 |                            |
                 |   Logical Device Table     |
                 |        Extension           |
                 |         (LDTX)             |
                 |                            |
                 +----------------------------+
```

Logical Device Table

Zero Entry Format

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
0|            Highest entry number               |
 +-----------------------------------------------+
1|               Entry size = 7                  |
 +-----------------------------------------------+
2|          Streams device number                |
 +-----------------------------------------------+
3|///////////////////////////////////////////////|
 +-----------------------------------------------+
4|///////////////////////////////////////////////|
 +-----------------------------------------------+
5|///////////////////////////////////////////////|
 +-----------------------------------------------+
6|///////////////////////////////////////////////|
 +-----------------------------------------------+
7|///////////////////////////////////////////////|
 +-----------------------------------------------+
```

Typical Entry Format

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
 |                File use count                 |0
 +-----------------------------------------------+
 | Volume table index if device type = 0-7, else |
 | main process pin # or spooler process pin #   |1
 +-----------------------+--+--+-----------------+
 |     Record width      |CS|FO|  Device type    |2
 +-----+--+--+--+--+--+--+--+--+-----------------+
 |Spool|Sy|Di|Dn|Tr|Hd|Cl|S |  Device-dependent |
 |state|st|ag|Rq|lr|r |as| Q|  info (see below) |3
 +-----+--+--+--+--+--+--+--+--------------------+
 |/////////|            XDD head index           |4
 +---------+-------------------------------------+
 |                CONTROL-Y pin                  |5
 +-----------------------------------------------+
 | Default output device -OR- default class index|6
 |              (see discussion)                 |
 +-----------------------------------------------+
```

```
Discussion:
  Word 2.(8:1) -- Communication system device if set.
  Word 2.(9:1) -- If set, there are special forms mounted on the device.
  Word 3.(0:2) -- Spooled state of the device:
                  0 -- Not spooled.
                  1 -- Owned by an input  spooler.
                  2 -- Owned by an output spooler.
  Word 3.(2:1) -- Device is available to system (not down).
  Word 3.(3:1) -- Device is available to diagnostics (obs).
  Word 3.(4:1) -- :DOWN requested, honored when use count = 0.
  Word 3.(5:1) -- If set, trailers are disabled.
  Word 3.(6:1) -- If set, headers are disabled. These two bits are
                  managed such that header/trailers are generated in
                  pairs or not at all.
  Word 3.(7:1) -- If I/O, word 6 is the Device Class Table
                  index/LDEV# of the default output class/device
                  associated with this device.
  Word 3.(8:1) -- Spooling has been enabled (spool queues  are
                  open) for this device.
  Word 3.(9:7) -- Device dependent information:
                  1.  For terminal-like devices,  the  default
                      terminal  type  to be used if not speci-
                      fied in the :HELLO command.
                  2.  For variable density tape drives:
     Word 3.(10:3) -- actual tape density.
     Word 3.(13:3) -- density requested in FOPEN for writes to
                      unlabelled tapes only.
                      For either:
                      0 = unknown density/no FOPEN w/ write.
                      1 = 1600 BPI
                      2 = 6250 BPI
                      3 = 800  BPI
```

Logical Device Table Extension (LDTX)

Overview of Data Segment

```
DST 14 (= %16)   +----------------------------+<-----DST %16
SIR 10 (= %12)   |                            |
                 |   Logical Device Table     |
                 |                            |
                 |          (LDT)             |
                 |                            |
                 +----------------------------+
                 |                            |
                 |   Logical Device Table     |
                 |        Extension           |
                 |         (LDTX)             |
                 |                            |
                 +----------------------------+
```

Zero Entry

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
  0 |            Highest entry number               |
    +-----------------------------------------------+
  1 |               Entry size = 5                  |
    +-----------------------------------------------+
  2 |///////////////////////////////////////////////|
    +-----------------------------------------------+
  3 |///////////////////////////////////////////////|
    +-----------------------------------------------+
  4 |///////////////////////////////////////////////|
    +-----------------------------------------------+
```

Typical entry

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
  0 | S|SD|CP|FS|DS| Reserved  |  Device-specific   |
    +--+--+--+--+--+-----------+                    |
  1 |                 information                   |
    +-------                                 -------+
  2 |                   fields.                     |
    +-------                                 -------+
  3 |         See the following examples            |
    +-------                                 -------+
  4 |               of LDTX entries.                |
    +-----------------------------------------------+
```

Where:

S.....Seek ahead enable/disable flag (system or PV disc only).
SD....This logical device is a Serial Disc or a Foreign Disc.
CP....This logical device uses the CIPER protocol.
FS....This is a system or PV disc with Disc Free Space management.
DS....This LDEV is a DS or data communications device.

Terminal Entry

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
  0 | 0| 0| 0| 0| 0|  Reserved   |      TBRC       |
    +--+--+--+--+--+-------------+-----------------+
  1 |       Terminal Descriptor Table Offset        |
    +--+--------------------------------------------+
  2 |WS|////////////////////////////////////////////|
    +--+--------------------------------------------+
  3 |///////////////////////////////////////////////|
    +-----------------------------------------------+
  4 |///////////////////////////////////////////////|
    +-----------------------------------------------+
```

TBRC..Terminal's baud rate code (CPS = characters per second).

Speed (CPS)	ADCC/ATP (HPIB) TBRC
Not known	0
1920	16 (ATP only)
960	8
480	9
240	7
120	11
60	6
30	13
15	14
14	---
10	15

WS....This terminal is connected to a Workstation Configurator port.

TDT offset...Offset from the base of the Terminal Descriptor Table (TDT) to the TDT entry for this terminal. A -1 indicates no TDT entry exists for this terminal.

Serial or Foreign Disc Entry

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
  0 | 0| 1| 0| 0| 0|   Reserved    |/////////////////|
    +--+--+--+--+--+---------------+-----------------+
  1 |      SDISC:  XDS# for variables, Gap Table      |
    |      FDISC:  1                                  |
    +-------------------------------------------------+
  2 |      SDISC:  1 ==> data buffer XDS's acquired   |
    |      FDISC:  not used.                          |
    +-------------------------------------------------+
  3 |      SDISC:  PCB index when WAITing, else 0     |
    |      FDISC:  not used.                          |
    +-------------------------------------------------+
  4 |/////////////////////////////////////////////////|
    +-------------------------------------------------+
```

CIPER Entry

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
  0 | 0| 0| 1| 0| 0| Reserved   |DB|/////////////////|
    +--+--+--+--+--+------------+--+-----------------+
  1 |  CIPER Device Control Data Segment # (CDCDS)    |
    +--+----------------------------------------------+
  2 |DN|       CTM Index for this device (CTMI)       |
    +--+----------------------------------------------+
  3 |/////////////////////////////////////////////////|
    +-------------------------------------------------+
  4 |/////////////////////////////////////////////////|
    +-------------------------------------------------+
```

DB.....If set to 1, then debugging is in effect.
DN.....If 1, the CIPER facility has been de-activated for this device because of error.
CTMI...Control Table Map Index (an index into the Control Table Map (CTM), which is located in the CDCDS.

System or Private Volume Disc Entry

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
  0 | S| 0| 0| 1| 0|   Reserved    |/////////////////|
    +--+--+--+--+--+---------------+-----------------+
  1 |/////////////////////////////////////////////////|
    +-------------------------------------------------+
  2 |       Disc Free Space DST number (DFSDST)       |
    +-------------------------------------------------+
  3 |      Disc Free Space error status (DFSERR)      |
    +-------------------------------------------------+
  4 |/////////////////////////////////////////////////|
    +-------------------------------------------------+
```

S......Seek ahead enable/disable flag.

Device Class Table (DCT)

Overview of Data Segment

```
DST 40 (= %50)    +---------------------------+<-----DST %50
SIR 40 (= %50)    |                           |
                  |     Device Class Table    |
                  |                           |
                  |           (DCT)           |
                  |                           |
                  +---------------------------+
                  |                           |
                  | Terminal Descriptor Table |
                  |                           |
                  |           (TDT)           |
                  |                           |
                  +---------------------------+
```

Device Class Table

Header Entry Format

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
  0|           Total table (segment) size           |
   +------------------------------------------------+
  1|  Entry size (variable, this word set to 1)     |
   +------------------------------------------------+
  2|         Number of device class entries         |
   +------------------------------------------------+
  3|      Pointer to first device class entry       |
   |               (segment relative)               |
   +------------------------------------------------+
  4|      Number of terminal descriptor entries     |
   +------------------------------------------------+
  5|  Pointer to first terminal descriptor entry    |
   |               (segment relative)               |
   +------------------------------------------------+
```

Typical Entry Format

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
    0|                                               |
     |                                               |
    1|                                               |
     |              Class name (ASCII)               |
    2|                                               |
     |                                               |
    3|                                               |
     +--+--------------------+--+--+-----------------+
    4|//|  Cyclical pointer  |SQ| T|Class Access Type|
     +--+--------------------+--+--+-----------------+
    5|        Number of devices in class (N)         |
     +-----------------------------------------------+
    6|                    LDEV #1                    |
     +-----------------------------------------------+
    7|                    LDEV #2                    |
     +-----------------------------------------------+
                 .                      .
                 .                      .
                 .                      .
     +-----------------------------------------------+
  N+5|                    LDEV # N                   |
     +-----------------------------------------------+
```

Discussion:

The Device Class Table (DCT) contains a varying number of variable length entries. This is because you may configure an arbitrary number of device classes on a system, and each device class may be comprised of an arbitrary number of logical devices. There is one DCT entry per device class, and each DCT entry contains a list of logical devices in the class. There is no established order of entries in the DCT, nor is there an order of LDEVs within an entry.

Due to the haphazard nature of the DCT, its overall properties are kept in the header entry. These include the segment-relative starting address of the DCT (in case the header entry should be expanded later) and the number of entries in the table A segment-relative pointer to the Terminal Descriptor Table (which follows the DCT) may also be used to calculate the size of the DCT. Also note the "Entry size" word. It is meaningless for this table, but is included for compatibility with other fixed-length entry MPE tables.
Since the DCT entries are of variable length, when you want a particular entry you must always start at the beginning of the DCT and link through each entry until you find the one you're interested in.

A few of the fields in the DCT require further description:

Word 4.(1:7) -- Cyclical pointer. Currently used only for system and private volume disc devices. The pointer varies from 1 to N (number of entries in the class) and indicates the LDEV# in the class list on which the last extent was allocated. The disc space allocation routines will try to satisfy the next re- quest on the next disc drive indicated by the cyclical pointer (with wraparound to 1 if the pointer > N). If that fails, the pointer is incremented until space is found or all devices in the class have been tried.

Word 4.(8:1) -- If set, spooling has been enabled (spool queues opened) for this device class.

Word 4.(9:1) -- If set, the class is a terminal type class.

Word 4.(10:6) -- Usually the same as the device type represented sented by the class (0 for24 for tape, 32 for printer, etc.). Serial disc classes are disc devices accessed as tape drives, so their true device types are kept in the LDT, while this field holds a special cial type (31, or %37), indicating a serial I/O (non-concurrent) device. Similarly, a foreign disc is a nonsharable disc drive, so that fact is reflected by a special type 7 in this field, even though the true hard- ware type is kept in the LDT, as for serial discs.

Interrupt Linkage Table (ILT) for HP-IB Systems

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
0   |          Channel                               |   ICPVA0
1   |           Program                              |   ICPVA01
2   |            Variable                            |   ICPVA02
3   |             Area   (ICPVA)                     |   ICPVA03
    +------------------------------------------------+
4   |          DMA Abort                             |   ICPVA04
5   |           Address                              |   ICPVA05
    +------------------------------------------------+
6   |                         0                      |   ISRQL/ICPGM
    +------------------------------------------------+
7   | M|     CHANQUE      |     |    CHAN    | DEV   |   ICNTRL
    +------------------------------------------------+
%10 |SYSDB relative pointer to channel program area.|   ISIOP
    +------------------------------------------------+
%11 |SYSDB relative pointer to status return area.  |   ISTAP
    +------------------------------------------------+
%12 |single instruction that is executed to extract |   IUNIT
    |the device unit number from the status pointed |
    |to by ISTAP.                                    |
    +------------------------------------------------+
%13 |SYSDB relative DIT pointer of the device        |   ICDP
    |currently using the channel to perform a data   |
    |operation.                                      |
    +------------------------------------------------+
%14 |        SIOPSIZE         |         CQUEN        |   IQUEUE
    +------------------------------------------------+
%15 |RW|WP|IG|SC|SQ|                    |   HCUNIT   |   IFLAG
    +------------------------------------------------+
%16 | SYSDB relative DIT pointer for unit 0          |   IDITP0
    +------------------------------------------------+
                          .
                          .
                          .
    +------------------------------------------------+
    |SYSDB relative DIT pointer for unit n           |   IDITPN
    +------------------------------------------------+
    |       Program status return area               |
    |        pointed to by ISTAP                     |
    +------------------------------------------------+
    |           Seekmask  (Disc only)                |
    +------------------------------------------------+
    |             I/O                                |
    |              Program                           |
    |               Area                             |
    +------------------------------------------------+
```

ILT (Cont.)

```
 IPCVA - These four words comprise the channel program
         variable area where information is stored concerning
         a channel program Interrupt instruction or abort.
         CPVA0 should be used only for channel program aborts.
ICPVA4 - Words 4 and 5 contain DMA address, when channel program
         aborts during DMA transfer.
ISRQL  - Serial poll request queue length.  HP-IB Systems do
         not support any serial poll devices.  This should
         always be zero.
ICPGM  - This is the SYSDB relative address of the channel program
         to be started for this device after receiving a HIOP
         interrupt in GIP.  GIP will call STARTIO when the flags
         word indicates "ignore halt interrupt" and "start channel
         program" bits are set.
ICNTRL - Contains controller information.
    .M   If set, the controller is sharing a software channel
         resource in order to limit bandwidth.
 .CHNQ   The software channel resource number.
 .DRTN   The DRT number for a Series 33 device is equivalent to:
          .CHAN - channel number (4 most significant bits of DRTN)
          .DEV  - device number (3 least significant bits of DRTN)
IFLAG  - Used for controller flags.
   .RW   Runwait flag.  An idle channel program should be started
         when there are no active requests to process.
   .WP   Waitprog flag. An idle channel program has been started
         for this controller.  This bit is reset by an interrupt.
   .IG   Ignorehi flag. An HIOP instruction has been issued against
         this controller, but the channel program was not in a
         wait statement.  Therefore, ignore the interrupt generated
         by the channel code when this program halts.
   .SC   Start channel program flag.  When set along with the IG
         flag, GIP will start a previously attempted SIOP on this
         device.
   .SQ   Start channel program "queued" flag.  When bit SC is set,
         this bit will determine if the call to START'HPIB will
         have logical parameter QUEUED true or false.
 .HCUNIT Highest configured unit number for this controller.
```

Device Information Table (DIT)

There is one DIT per physical device. If a physical device represents represents more than one logical device, the logical device number is obtained from the I/O queue element. Although details of DIT's vary with device, the following structure is common to all:

DIT for HP-IB Systems

```
   0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
0 |T |D |AC|RQ|SI|MU| 0|IO|IA|NO|ST|NS|  STATE    |   DFLAG
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
1 |SYSDB relative pointer to the DIT for the next |   DLINK
  |   device requesting this resource or service  |
  +-----------------------------------------------+
2 |SYSDB relative pointer to the first IOQ in     |   DIOQP
  | request list for this device                  |
  +-----------------------------------------------+
3 |            Logical device humber              |   DLDEV
  +-----------------------------------------------+
4 |SYSDB relative pointer to Device Linkage Table |   DDLTP
  +-----------------------------------------------+
5 |SYSDB relative pntr to Interrupt Linkage Table |   DILTP
  +-----------------------------------------------+
6 |      Controller Hardware Status               |   DSTAT
  +-----------------------------------------------+
7 |Hardware error status.  Set when the driver    |   DSERR
  | detects an error.  Whenever <>0, the driver   |
  | monitor logs an I/O error and clears this word|
  +-----------------------------------------------+
8 |        Device Dependent Area                  |   (DTIME)
  +-----------------------------------------------+
9 |        Device Dependent Area                  |   (DTRQX)
  +-----------------------------------------------+
10| IOT |//////////////////|     Phys. unit #     |   DUNIT
  +-----------------------------------------------+
```

DTRQX   Used by some device drivers, it denotes timer request index.

DIT Terminology for HP-IB Systems

```
 DFLAG - DEVICE RELATIVE FLAGS
  T     SET IF DEVICE IS A TERMINAL.
  D     SET IF DEVICE IS A DISC.
  AC    ACTIVE BIT. 1 IMPLIES A MONITOR CURRENTLY SERVICING
        THIS DEVICE.
  RQ    REQUEST BIT. 1 IMPLIES SERVICE REQUESTED WHILE
        MONITOR IS ACTIVE.
  MU    IF SET, MULTIPLE UNIT CONTROLLER.
  IO    IF SET, THEN A CHANNEL PROGRAM IS CURRENTLY EXECUTING.
  IA    IF SET, AN INTERRUPT OR RESPONSE HAS OCCURRED.
  NO    IF SET, DEVICE IS IN A NOT READY OR OPERATOR WAIT.
  ST    IF SET, AN IDLE CHANNEL PROGRAM SHOULD BE STARTED FOR
        THIS DEVICE.
  SI    SPECIAL INTERRUPT HANDLER
  NS    DO NOT SHORT WAIT THIS DISC
  STATE CURRENT DRIVER STATE AS DEFINED BY THE MONITOR.
        ALLOWABLE STATES ARE:
         0 - START REQUEST
         1 - NOT USED (BUT RESERVED)
         2 - CALL DRIVER INITIATOR
         3 - CALL DRIVER COMPLETOR
         4 - NOT USED (BUT RESERVED)
         5 - COMPLETE REQUEST
         6 - UNEXPECTED INTERRUPT OCCURRED
         7 - START OPERATOR INTERVENTION WAIT
       %10 - WAITING (ON OPERATOR). RESTART AT 0
       %11 - WAITING (DATA MAKEPRESENT/FREEZING)
       %12 - WAITING (INITIATOR CODE MAKEPRESENT/FREEZE)
       %13 - WAITING (FOR COMPLETION INTERRUPT)
       %14 - WAITING (FOR DEVICE CONTROLLER AVAILABILITY)
       %15 - NOT USED (BUT RESERVED)
       %16 - WAITING (INITIATOR CODE MAKEPRESENT)
       %17 - WAITING (COMPLETOR CODE MAKEPRESENT)
IOT - I/O System type  0-Series II/III I/O System
                       1-HP-IB Systems
                       2-unused
                       3-unused
```

Device Information Table (DIT) for CIPER

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element (however, this driver only supports one device per controller.) The following diagram shows the DIT used for the HP-IB CIPER physical driver.

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
   +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
  0| 0| 0|AC|RQ| 0| 0| 0|IO|IA|NO|ST| 0|   STATE   |  DFLAG
   +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
  1| SYSDB relative pointer to the DIT for the next|  DLINK
   | device requesting this resource or service    |
   +-----------------------------------------------+
  2| IOQ table index to the first IOQ in           |  DIOQP
   | request list for this device                  |
   +-----+----------------+------------------------+
  3| IOT |   Phys. unit # | Logical device number  |  DLDEV
   +-----+----------------+------------------------+
  4| SYSDB relative pointer to Device Linkage Table|  DDLTP
   +-----------------------------------------------+
  5| SYSDB relative pointer to Intrp Linkage Table |  DILTP
   +--+--+--+--+--+-------+------------------------+
  6|VS|AB|RE|TP|NR| NR CNT |     DEVICE STATUS     |  DSAVE
   +--+--+--+--+--+-------+------------------------+
  7| Hardware error status.  Set when the driver   |  DSERR
   | detects an error.  Whenever <>0, the driver   |
   | monitor logs an I/O error and clears this word|
   +-----------------------------------------------+
%10| Bit 0 is set at completion of timer           |  DTIME
   +-----------------------------------------------+
%11| Holds the time out request entry index while  |  DRQST
   | a timer is active.                            |
   +--+--+--+--+--------+--------+--------+--------+
%12|RF|UE|DE|TO|UNIT CNT|DATA CNT| TO CNT |PRTY CNT|  DCOUNTS
   +--+--+--+--+--------+--------+--------+--------+
%13|           Error logging location #1           | DLOGERROR
   +-----------------------------------------------+
%14|           Error logging location #2           | DLOGCOUNT
   +-----------------------------------------------+
```

```
DFLAG - Flags and request state
  AC  ACTIVE  - A monitor is currently servicing this device.
  RQ  REQUEST - A service request is pending while the monitor is
                active.
  IO  IOPROG  - An I/O Channel Program is running for this device.
  IA  IAK     - An interrupt or response has occurred for this device.
  NO  NOTRDY  - Go to state %10 after Idle Channel Program is started.
  ST  STWAIT  - The device monitor is starting an Idle Channel Program
                for this device.  There is no IOQ associated with this
                type of request.
  STATE       - State of the device monitor.  Specifies the next action
                to be taken in SIODM in servicing the request:
                  0 - start new request
                  1 - not used
                  2 - call driver initiator procedure
                  3 - call driver completor procedure
                  4 - not used
                  5 - process request completed
                  6 - initiate device recognition sequence
                  7 - start operator intervention wait
```

```
                %10 - wait for interrupt (operator intervention)
                      restart at state 0
                %11 - wait for data segment freeze, then state 2
                %12 - wait for driver initiator to be frozen, then
                      allocate controller (state 2)
                %13 - wait for I/O completion interrupt, then state 3
                %14 - wait for controller, then call driver initiator
                %15 - not used
                %16 - wait for initiator make present, then state 2
                %17 - wait for completor make present, then state 3

DLDEV - I/O system type, unit and logical device number
    0 - HP3000 Series iII/III
    1 - HP 3000 HP-IB
    2 - Unused
    3 - Unused

DSAVE - Device processing flags
  VS - VALID STATUS - Set to indicate Device Status has been updated.
  AB - DVRABFLAG    - Sequence Abort in progress due to ABORT request.
  RE - RETRYFLAG    - Sequence Abort in progress due to an error.
  TP - TIMERPOPPED  - Current error is due to software timer popping.
  NR - NOTRDYFLAG   - Not Ready Wait in progress.
  NR CNT            - Number of Not Ready Waits during this request.
  DEVICE STATUS     - Device status returned during a Sequence Abort.
       BIT  8       -   CRC available and enabled.
         "   9      -   Reserved.
         "  10      -   Reserved.
         "  11      -   Reserved.
         "  12      -   Power fail or reset has occurred.
         "  13      -   A protocol error has been detected.
         "  14      -   A parity error has been detected.
         "  15      -   The peripheral has data to send.

DSERR - Pointer to status to be logged.
        Bits(0:8)   - Number of words to be logged.
        Bits(8:8)   - Offset relative to DITP(0).

DCOUNTS             - Error flags and error counts (4).
   RF - REQ FAILED  - An error has forced this request to be aborted.
   UE - UNIT ERROR  - The current error is a Unit Error.
   DE - DATA ERROR  - The current error is a Data Error.
   TO - TIME OUT    - The current error is a GIC Time Out Error.
   UNIT CNT         - Number of Unit Errors during this request.
   DATA CNT         - Number of Data Errors during this request.
   TO CNT           - Number of GIC Time Outs during this request.
   PRTY CNT         - Number of HP-IB Parity Errors during this request.
```

DIT for Channel Devices

```
    0    1    2   3   4    5    6     7    8    9  10 11 12 13 14 15
  |-----------------------------------------------------------------|
 0|TERM|DISC|ACT|REQ|    | M  | SIO  | IO |IAK|  M |NT|  |   STATE   |DFLAG
  |    |    |   |   |    |UNIT| PREMP|PROG|   |HEAD|RY|  |           |
  |-----------------------------------------------------------------|
 1|                           NEXT DITP                             |DLINK
  |-----------------------------------------------------------------|
 2|                              IOQP                               |DIOQP
  |-----------------------------------------------------------------|
 3|                      LOGICAL DEVICE NUMBER                      |DLDEV
  |-----------------------------------------------------------------|
 4|                              DLTP                               |DLTP
  |-----------------------------------------------------------------|
 5|                              ILTP                               |DILTP
  |-----------------------------------------------------------------|
 6|                   Controller Hardware Status                    |DSTAT
  |-----------------------------------------------------------------|
 7|                     Hardware Error Status                       |DSERR
  |-----------------------------------------------------------------|
 8|                                                                 |DTIME
  |-----------------------------------------------------------------|
 9|                                                                 |DTRQX
  |-----------------------------------------------------------------|
10|   IOT  |                          |        PHYS. UNIT #         |DUNIT
  |-----------------------------------------------------------------|
  |                                                                 |


  |                    DRIVER DEPENDENT DIT AREA                    |
  |                                                                 |
  |                                                                 |
  |                                                                 |
   -----------------------------------------------------------------
```

```
DFLAG.TERMINAL - Device is a terminal
     .DISC     - Device is a Disc (Bit 0 = 0)
     .ACTIVE   - A monitor is currently servicing this device
     .REQUEST  - Service requested while monitor was active

     .MUNIT    - device controller servicing multiple units
     .SIOPREMPT- If set then a               request has been queued for
                 this device.  Preempt code is set in IOQ.
     .IOPROG   - I/O program in progress.  Decrement SIOCOUNT and
                 check for multi-channel when complete
     .IAK      - Interrupt or Response has occurred.
     .M HEAD   -Moving head disc
     .NT RDY   -Not ready for SIO. SIODM holds off next SIO until
                ALLOWPOLL is done.

DTRQX       - Used by some device drivers, it denotes timer
              request index.
```

DIT for Channel Devices (Cont.)

DFLAG.STATE - this quantity specifies the next action to be taken in servicing the request.

0-new - start request.
1-not used.
2-call Driver Initiator Procedure
3-call Driver Completor Procedure
5-complete request
6-device recognition
7-start operator intervention wait (%10)
%10-restart request on interrupt
%11-wait for data to be frozen then state 2
%12-wait for driver code to be frozen then state 2
%13-call completor on interrupt
%14-wait for device controller
%15-not used
%16-wait for initiator make present then state 2
%17-wait for completor make present then state 3

DLINK - SYSDB relative pointer to the DIT for the next device requesting this resource or service.
DIOQP - SYSDB relative pointer to the first IOQ in the request list for this device
DLDEV.LDEVN - Logical Device Number
.UNIT - unit number of the physical device.
.IOT - IO type 0=> Series III I/O, 1=> HPIB I/O
DDLTP - SYSDB relative pointer to the DLT.
DILTP - SYSDB relative pointer to the ILT.
DSTAT - interrupt status for this device. Set each time the device interrupts.
DSERR - Hardware Device Controller Status. Set when the driver detects an error. Whenever not zero, SIODB logs an I/O error and clears this word.
DTIME - time out completed flags. If a timeout occurs in response to a timer request type %20 (I/O request), the sign bit is set in this word. The IA bit in DFLAG is also set, and the monitor for this device is awakened. (Only used if timer services are requested. Must be word #8 if timer services are requested.)

DIT For 7905/7906/7920/7925

```
   0  1  2   3   4   5     6  7     8   9  10 11 12       15
  |--|--|---|---|---|-----|--|----|---|---|--|--|----------|
 0| 0| 1|ACT|REQ| CD|  M  | 0| I/O|IAK| 1 | 0| 0|  STATE   | 0  DFLAG
  |  |  |   |   |   |UNIT |  |PROG|   |   |  |  |          |
  |-------------------------------------------------------|
 1|                     NEXT DITP                         | 1  DLINK
  |-------------------------------------------------------|
 2|             CURRENT (ACTIVE) DISC REQUEST             |2   DIOQP
  |-------------------------------------------------------|
 3|                LOGICAL DEVICE NUMBER                  | 3  DLDEV
  |-------------------------------------------------------|
 4|                        DLTP                           | 4  DDLTP
  |-------------------------------------------------------|
 5|                        ILTP                           | 5  DILTP
  |-------------------------------------------------------|
 6|                 -1 WHEN POWER FAIL                    | 6  DRQST
  |-------------------------------------------------------|
 7| # OF ERROR WORDS TO LOG    |  DIT REL ADDR TO LOG     | 7  DSERR
  |-------------------------------------------------------|
 8|         INDEX OF FIRST REQUEST IN QUEUE               |10  DMAMQ
  |-------------------------------------------------------|
 9|         INDEX OF LAST REQUEST IN QUEUE                |11  DMAMQT
  |-------------------------------------------------------|
10| IOT |/////////////////////////|    PHYSICAL UNIT #    |12  DUNIT
  |-------------------------------------------------------|
11|        SIO PROGRAM-RELATIVE ABORT ADDRESS             |13  DLOGSIOP
  |-------------------------------------------------------|
12|                  CURRENT PHYSICAL                     |14  CPDA
13|                   DISK ADDRESS                        |15
  |-------------------------------------------------------|
14|              CURRENT DATA BUFFER ADDRESS              |16  CDBA
  |-------------------------------------------------------|
15|                WORD COUNT REMAINING                   |17  WCR
  |-------------------------------------------------------|
16|                 CURRENT WORD COUNT                    |20  CWC
  |-------------------------------------------------------|
17|                   SYSBUF INDEX                        |21  SYSBUFA
  |-------------------------------------------------------|
18|                  STATUS 1 RETURN                      |22  STAT1
  |-------------------------------------------------------|
19|                  STATUS 2 RETURN                      |23  STAT2
  |-------------------------------------------------------|
20|                        CYL                            |24  CEDA
  |-------------------------------------------------------|
21|            HEAD         |         SECTOR              |25
  |-------------------------------------------------------|  \
22|                  STATUS 1 RETURN                      |   |
  |-------------------------------------------------------|   |
23|                        CYL                            |   |
  |-------------------------------------------------------|   |
```

DIT for 7905/7906/7920/7925 (Cont.)

```
  |-------------------------------------------------------|   |
24|            HEAD         |         SECTOR              |   |
  |-------------------------------------------------------|  REQUEST
25|                   DISPLACEMENT                        |  SYNDROME
  |-------------------------------------------------------|   |
26|                      PATT 1                           |   |
  |-------------------------------------------------------|   |
27|                      PATT 2                           |   |
  |-------------------------------------------------------|   |
28|                      PATT 3                           |   |
  |-------------------------------------------------------|  /
29|              SECTOR COUNT TO TRANSFER                 |35  SCOUNT
  |-------------------------------------------------------|
30|                                                       |36  INITADR
  |                 INITIALIZE ADDRESS                    |
31|                                                       |37
  |-------------------------------------------------------|
32|                                                   | L |40  DMISC
  |-------------------------------------------------------|
33|              CNTLR STATUS AFTER SEEK                  |41  SEEKSTAT
34|               IN CHANNEL PROGRAM                      |42
  |-------------------------------------------------------|
35|            CPVA WORD 0 UPON CHANNEL ABORT             |43  DLOGERROR
  |-------------------------------------------------------|
36|            CURRENT LOGICAL SECTOR ADDRESS             |44  CLDA
  |-------------------------------------------------------|
  |                                                       |
  |                                                       |
  |_______________________________________________________|
```

DMISC
(15:1) L'STAT'ERR - 1 Last transfer ended in error.

IOT - I/O Devices
0 - non-HP-IB
1 - HP-IB Systems
2 - unused
3 - unused

Error and Retry Information

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 | D| S| E| M| W| T| O| C|CL| O  O  O  O|retry cnt     QMISC OF IOQ
 |--|--|--|--|--|--|--|--|--|-----------|---------|
```

D - retry determination
S - request syndrome
E - request error information
M - update track map
W - writing track map
C - issued a recalibration
CL- driver issuing channel clear
T - timeout wait

NOTE: Integrated Cartridge Tape's DIT has the same format.

CS 80 Disc Device Information Table (DIT)

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element. For the CS'80 disc controller, there will only be one device. The following diagram shows the DIT used by the CS'80 disc driver.

```
0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15  MNEMONIC
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+
0|TM|DS|AC|RQ|CD| 0| 0|IO|IA|NO|ST| 0|   STATE   | DFLAG
   +--+--+--+--+--+--+--+--+--+--+--+-----------+
1| SYSDB relative pointer to the DIT for the next| DLINK | device requesting
this resource or service    |
   +-----------------------------------------------+
2|             Current request index             | DCURREQP
   +-----------------------------------------------+
3|             Logical device number             | DLDEV
   +-----------------------------------------------+
4| SYSDB relative pointer to Device Linkage Table| DDLTP
   +-----------------------------------------------+
5| SYSDB relative pointer to Intrp Linkage Table | DILTP
   +-----------------------------------------------+
6| DSTAT is -1 when a system powerfail occurred  | DSTAT
   +-----------------------------------------------+
7| Hardware error status.  Set when the driver   | DSERR | detects an error.
Whenever <>0, the driver   |  | monitor logs an I/O error and clears this
word|
   +-----------------------------------------------+
%10|         index of first request in queue       | DQHEAD *
   +-----------------------------------------------+
%11|         index of last request in queue        | DQTAIL *
   +-----+-----------------+-----------------------+
%12| IOT |    Physical  Unit  #                    | DUNIT
   |--+--+-----------------+-----------+-----------+
%13| Table relative index to system buffer element |DSBUFADDR
   +-----------------------------------------------+
%14| High order logical sector address of bad blk  | DBADBLK1
   +-----------------------------------------------+
%15| Low order logical sector address of bad blk   | DBADBLK2
   +-----------------------------------------------+
%16| Byte transfer left when bad block occurred    | DBADXFER
   +-----------------------------------------------+
%17| Hardware logged error status - CPVA (0)       | DLOGERROR
   +-----------------------------------------------+
%20| Channel program aborted relative offset       | DSIOPSTOP
   +-----------------------------------------------+
%21| Disc status (20 bytes)-Logged on status error | DSTATUS
   +-----------------------------------------------+
 . |                      .                                             |
   +-----------------------------------------------+
 . |                      .                                             |
   +-----------------------------------------------+
%33|LK|IF|MD|                       | SUBSTATE  | DMISC
```

```
   +--+--+--+--+-----------------------------------+
%34|RE|DC|DR|EN|                       |LOCAL STATE| RPSWORD1
   +--+--+--+--+-----------------------+-----------+
%35|          T1           |           T2          | RPSWORD2
   +-----------------------+-----------------------+
```

DFLAG - Flags and request state

TM TERM - Set if device is a terminal. DS DISC - If TM = 0 and this bit is set then the device is a disc, otherwise device dependent. AC ACTIVE - A monitor is currently servicing this device. RQ REQUEST - A service request is pending while the monitor is active. IO IOPROG - An I/O Channel Program is running for this device. IA IAK - An interrupt or response has occurred for this device. NO NOTRDY - Go to state %10 after Idle Channel Program is started. ST STWAIT - The device monitor is starting an Idle Channel Program for this device. There is no IOQ associated with this type of request. STATE - State of the device monitor. Specifies the next action to be taken in SIODM in servicing the request:

0 - start new request 1 - not used 2 - call driver initiator procedure 3 - call driver completor procedure 4 - not used 5 - process request completed 6 - initiate device recognition sequence 7 - start operator intervention wait %10 - wait for interrupt (operator intervention) restart at state 0 %11 - wait for data segment freeze, then state 2 %12 - wait for driver initiator to be frozen, then allocate controller (state 2) %13 - wait for I/O completion interrupt, then state 3 %14 - wait for controller, then call driver initiator %15 - not used %16 - wait for initiator make present, then state 2 %17 - wait for completor make present, then state 3

DLINK - A SYSDB relative pointer to the next DIT requesting this resource or service.

DCURREQP - A current request sysbase index.

DUNIT.(0:2) - I/O system type

0 - non-HP-IB 1 - HP3000 HP-IB Systems 2 - Unused 3 - Unused

DLDEV - Logical device number of this device.

DSTAT - Set to a -1 when a system powerfail has occurred.

DSERR - Pointer to status to be logged.

Bits(0:7) - Number of words to be logged. Bits(8:15) - Offset relative to DITP(0).

DMISC - Device dependent processing flags

LOCK'FLG - Lock flag denoting unload status of the disc volume.

0 - Allow operator unload to the volume. 1 - Deny operator unload to the volume.

IGNORE'INT'FLG - Ignore unexpected interrupt flag.

SUBSTATE - Indicates state of the idle channel program:

0 - Normal idle channel program wait 1 - Idle request being serviced wait

DSBUFADDR - SYSDB relative pointer to the system buffer element used to read the DSCT. Zero, if no element gotten.

DBADBLK1 - High order logical sector address of the bad block for the Defective Sector Table (DSCT) entry.

DBADBLK2 - Low order logical sector address of the bad block for the DSCT entry.

DBADXFER - Byte transfer left when bad block occurred.

DLOGERROR - CPVA(0) logged on hardware error status.

DSIOPSTOP - Stopped channel program relative offset location due to an error in CPVA(0).

DSTATUS - 20 bytes disc status logged on status error. (See CS'80 Disc Drive Status).

RPSWORD1 - Flags and local state

RE - Read revision code done. Set if read revision code level is done. DC - RPS revision code. Set if controller is "PEP"ed. DR - RPS desirable. Set if RPS is desirable. EN - RPS enabled. Set if default value for RPS is enabled. MR - Driver is processing a marginal data error from the drive. Do not return hard error. Local State - State of the local request made by driver

0 - No local request is being processed 1 - Reading rev code 2 - Setting default RPS

RPSWORD2 - Default value for RPS

T1 - Time to target in hundreds of microseconds T2 - Window size in hundreds of microseconds

DIT For 7970 Magnetic Tape

```
    0  1  2   3   4   5    6   7    8   9  10 11 12       15
  |--|--|---|---|---|-----|--|----|---|---|--|--|----------|
  | 0| 0|ACT|REQ| 0 |  M  | 0| I/O|IAK| 0 | 0| 0|  STATE   |    DFLAG
  |  |  |   |   |   |UNIT |  |PROG|   |   |  |  |          |
  |--------------------------------------------------------|
 1|                      NEXT DITP                         |    DLINK
  |--------------------------------------------------------|
 2|                        IOQP                            |    DIOQP
  |--------------------------------------------------------|
 3|                LOGICAL DEVICE NUMBER                   |    DLDEV
  |--------------------------------------------------------|
 4|                      DLT PTR                           |    DDLTP
  |--------------------------------------------------------|
 5|                      ILT PTR                           |    DILTP
  |--------------------------------------------------------|
 6|RW|RU|SH|CE|DC|       HARDWARE STATUS                   |    DSTAT
  |--------------------------------------------------------|
 7|                    ERROR STATUS                        |    DSERR
  |--------------------------------------------------------|
 8|                   TIMEOUT FLAGS                        |    DTIME
  |--------------------------------------------------------|
 9|                TIMER REQUEST INDEX                     |    DTRQX
  |--------------------------------------------------------|
10| IOT |///////////////////////////|   PHYSICAL UNIT #    |    DUNIT
  |--------------------------------------------------------|
11|                                          13|RB4| RW    |    DDFLAGS
  |--------------------------------------------------------|
```

IOT - I/O Devices
0 - non-HP-IB
1 - HP-IB Systems
3 - unused
4 - unused

DSAVE - Device processing flags

RW RWBIT - Indicates tape has been rewound.

RU RWUNLD - Indicates that a rewind/unload was performed to allow a write-ring mount.

SH SHORT - A short read is in progress. After completion of read, EOF is checked for and if not present, the requested bytes are transferred from the short-read buffer to the user's buffer.

CE CESTAT - Channel parity error processing is in progress.

DC DSFLAG - Transfer used data chaining - used for computing the transmission log.

RW - (DDFLAGS, bit 15) if set, tape is rewound

RB4 - (bit 14) if set, need to rewind tape before next write

```
                     QMISC

 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15

|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  | FORWARD|  BACK  |        |
| R| B| F| G| E| S| U| SPACE  | SPACE  | RETRY  |
|  |  |  |  |  |  |  |COUNTER |COUNTER |COUNTER |
|--|--|--|--|--|--|--|--------|--------|--------|
```

Where
   R - retry in progress
   B - backspace in progress
   F - forward space in progress
   G - gap in progress
   E - backspace on data end-of-file
   S - short read in progress
   U - unload tape for write ring installation

DIT for 7976 Magnetic Tape

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element. The following diagram shows the DIT used for the mag tape driver.

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
   +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
  0| 0| 0|AC|RQ| 0|MU| 0|IO|IA| 0| 0| 0|   STATE   |   DFLAG
   +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
  1| SYSDB relative pointer to the DIT for the next|   DLINK
   | device requesting this resource or service    |
   +-----------------------------------------------+
  2| SYSDB relative pointer to the first IOQ in    |   DIOQP
   | request list for this device                  |
   +-----------------------------------------------+
  3|             Logical device number             |   DLDEV
   +-----------------------------------------------+
  4| SYSDB relative pointer to Device Linkage Table|   DDLTP
   +-----------------------------------------------+
  5| SYSDB relative pntr to Interrupt Linkage Table|   DILTP
   +-----------------------------------------------+
  6|RW|RU|SH|  |DC|PF|                             |   DSAVE
   +-----------------------------------------------+
  7| Hardware error status.  Set when the driver   |   DSERR
   | detects an error.  Whenever <>0, the driver   |
   | monitor logs an I/O error and clears this word|
   +--+--+--+--+-----------------------------------+
%10| Bit 0 is set at completion of timer           |   DTIME
   +--+--+--+--+-----------------------------------+
%11| Interrupt status for this unit.  Set by the   |   DSTAT
   | driver each time it processes an interrupt.   |
   +-----------------------------------------------+
%12| IOT |//////////////////|   Physical unit #    |
   |-----------------------------------------------|
%13| Holds the time out request entry index while  |   DRQST
   | a timer is active.                            |
   +-----------------------------------------------+
%14| Error log. Contains 5 valid bytes of status   |   DLOGERROR
   +-----------------------------------------------+
```

DFLAG - Flags and request state
  AC ACTIVE - A monitor is currently servicing this device.
  RQ REQUEST - A service request is pending while the monitor is active.
  MU MUNIT   - This device is on a multi-unit controller.
  IO IOPROG - An I/O Channel Program is running for this device.
  IA IAK   - An interrupt or response has occurred for this device.
  NO NOTRDY - Go to state %10 after Idle Channel Program is started.
  ST STWAIT - The device monitor is starting an Idle Channel Program for this device. There is no IOQ associated with this type of request.

STATE - State of the device monitor. Specifies the next action to be taken in SIODM in servicing the request:
   0 - start new request
   1 - not used
   2 - call driver initiator procedure
   3 - call driver completor procedure
   4 - not used
   5 - process request completed
   6 - initiate device recognition sequence
   7 - start operator intervention wait
   %10 - wait for interrupt (operator intervention) restart at state 0
   %11 - wait for data segment freeze, then state 2
   %12 - wait for driver initiator to be frozen, then allocate controller (state 2)
   %13 - wait for I/O completion interrupt, then state 3
   %14 - wait for controller, then call driver initiator
   %15 - not used
   %16 - wait for initiator make present, then state 2
   %17 - wait for completor make present, then state 3

DSAVE - Device processing flags
  RW RWBIT - Indicates tape has been rewound.
  RU RWUNLD - Indicates that a rewind/unload was performed to allow a write-ring mount.
  SH SHORT - A short read is in progress. After completion of read, EOF is checked for and if not present, the requested bytes are transferred from the short-read buffer to the user's buffer.

  DC DSFLAG - Transfer used data chaining - used for computing the transmission log.
  PF POWER - Device power up indication.

DSTAT - Mag tape controller status

BITS	USE
0	END OF FILE (EOF)
1	BEGINNING OF TAPE (BOT) / LOAD POINT (LP)
2	END OF TAPE (EOT)
3	SINGLE TRACK ERROR (NOT LOGGED FOR READS)
4	COMMAND REJECT (REJECT)
5	FILE PROTECT (NOT WRITE ENABLED; NO WRITE RING)
6	MULTIPLE TRACK ERROR (MTE)
7	UNIT ONLINE
8	GCR (6250 BPI DENSITY)
9	UNIT NUMBER (MSB)
10	UNIT NUMBER (LSB)
11	TIMING ERROR
12	TAPE RUNAWAY
13	REWINDING *
14	UNIT BUSY ** (REPORTED AS UNIT NOT READY)
15	INTERFACE BUSY *

Card Reader DIT

```
      0  1  2   3   4   5    6   7    8   9    10 11  12       15
     |--|--|---|---|---|-----|--|----|---|----|---|---|----------|
     | 0| 0|ACT|REQ| 0 |  0  |  | I/O|IAK|READ| NR|   | MSTATE   |  DFLAG
     |  |  |   |   |   |     |  |PROG|   |DONE|MSG|   |          |
     |-------------------------------------------------------------|
    1|                  DITP LINK TO NEXT DIT                       |  DLINK
     |-------------------------------------------------------------|
    2|                IOQP POINTER TO 1st REQUEST                   |  DIOQP
     |-------------------------------------------------------------|
    3|               LOGICAL DEVICE NUMBER                          |  DLDEV
     |-------------------------------------------------------------|
    4|               DRIVER LINKAGE TABLE POINTER                   |  DDLTP
     |-------------------------------------------------------------|
    5|               INTERRUPT LINKAGE TABLE POINTER                |  DILTP
     |-------------------------------------------------------------|
    6|                      (SEE BELOW)                             |  DSTAT
     |-------------------------------------------------------------|
    7|                  ERROR STATUS IF NOT 0                       |  DSERR
     |-------------------------------------------------------------|
  %10|                  REQUESTED WORD COUNT                        |  DTIME
     |-------------------------------------------------------------|
  %11|//////////////////////////////////////////////////////////////|  DTRQX
     |-------------------------------------------------------------|
  %12| IOT |///////////////////////////|     PHYSICAL UNIT #      |  DUNIT
     |-------------------------------------------------------------|
```

DSTAT bits:

```
BIT0=SIO OK
BIT1=0
BIT2=INT PENDING
BIT3=TIMING ERROR
BIT4=LIGHT DARK CHECK
BITS 5-6 =   00 COLUMN BINARY MODE
             01 UNUSED
             10 PACKED BINARY MODE
             11 HOLLERITH-TO-ASCII MODE
BIT7=COMPARE ERROR
BIT8=EOF DETECTED
BITS 9-10 =  00 NORMAL
             01 HOPPER EMPTY
             10 UNUSED
             11 STACKER FULL
BIT11=INVALID HOLLERITH
BIT12=PICK FAIL OR MOTOR CHECK
BIT13=TEST
BIT14=TROUBLE
BIT15=NOT READY
```

Card Reader DIT Field Definitions

DFLAG - Flags and device state

```
ACTIVE      Monitor is currently active servicing this device.

REQUEST   Service for this device was requested while the monitor was active.

IOPROG      SIO program in progress.

IAK         Interrupt occurred or request aborted or preempted.

READDONE    Previous read resulted in an EOF with a backup save
            requested.  The data has been saved  in an auxiliary
            buffer and will be passed back on the next read request.

NRMESSAGE   Set when a not ready message has been issued, and cleared
            when the reader is found ready.  Used to prevent multiple
            Not Ready messages when power is turned on.

MSTATE      Monitor State.  See SIODM specifications for details.
```

DLINK - SYSDB relative pointer to the DIT for the next device requesting service for this resource.

DIOQP - SYSDB relative pointer to the first IOQ element in the request list for this device.

DLDEV - Logical device number and unit number.

```
UNIT        Unit number of device.

LDEVN       Logical device number.
```

DDLTP - SYSDB relative pointer to driver linkage table (DLT).

DSTAT - Device interrupt status. Contains the device interrupt status at the last interrupt. See hardware ERS for details.
DSERR - Device interrupt error status. If not zero, then holds the device interrupt status from an operation with an erroneous completion status. Causes SIODM to log an error.

DWCNT - Holds the requested transfer count in words.

Device Information Table for HP-IB Card Reader

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element. The following diagram shows the DIT used for the card reader driver.

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
   +--+--+--+--+--+--+--+--+--+--+--+--+--------------+
  0| 0| 0|AC|RQ| 0|MU| 0|IO|IA|NO|ST| 0|    STATE     |   DFLAG
   +--+--+--+--+--+--+--+--+--+--+--+--+--------------+
  1| SYSDB relative pointer to the DIT for the next|   DLINK
   | device requesting this resource or service     |
   +-----------------------------------------------+
  2| IOQ table relative index to the first IOQ in  |   DIOQP
   | request list for this device                   |
   +--------+----------+----------------------------+
  3|            Logical device number              |   DLDEV
   +--------+----------+----------------------------+
  4| SYSDB relative pointer to Device Linkage Table|   DDLTP
   +-----------------------------------------------+
  5| SYSDB relative pntr to Interrupt Linkage Table|   DILTP
   +-----------------------------------------------+
  6|RD|AF|                                        |   DSAVE
   +-----------------------------------------------+
  7| Hardware error status.  Set when the driver   |   DSERR
   | detects an error.  Whenever <>0, the driver    |
   | monitor logs an I/O error and clears this word|
   +--+--+--+--+--------------------------------------+
%10| Not Used                                      |   DTIME
   +--+--+--+--+--------------------------------------+
%11|          Request word count                   |   DWCNT
   +-----------------------------------------------+
%12| IOT |/////////////////|      Physical unit #  |   DUNIT
   |-----------------------------------------------|
%13| Device Status.  Read from device during       |   DSTAT
   | each execution of the channel program.         |
   +-----------------------------------------------+
%14! Logging will be done from here.               !   DLOGERROR
   +-----------------------------------------------+
```

```
DFLAG - Flags and request state
  AC  ACTIVE  - A monitor is currently servicing this device.
  RQ  REQUEST - A service request is pending while the monitor is active.
  MU  MUNIT   - This device is on a multi-unit controller.
  IO  IOPROG  - An I/O Channel Program is running for this device.
  IA  IAK     - An interrupt or response has occurred for this device.
  NO  NOTRDY  - Go to state %10 after Idle Channel Program is started.
  ST  STWAIT  - The device monitor is starting an Idle Channel Program
                for this device.  There is no IOQ associated with this
                type of request.
```

```
  STATE       - State of the device monitor.  Specifies the next action
                to be taken in SIODM in servicing the request:
                  0 - start new request
                  1 - not used
                  2 - call driver initiator procedure
                  3 - call driver completor procedure
                  4 - not used
                  5 - process request completed
                  6 - initiate device recognition sequence
                  7 - start operator intervention wait
                %10 - wait for interrupt (operator intervention)
                      restart at state 0
                %11 - wait for data segment freeze, then state 2
                %12 - wait for driver initiator to be frozen, then
                      allocate controller (state 2)
                %13 - wait for I/O completion interrupt, then state 3
                %14 - wait for controller, then call driver initiator
                %15 - not used
                %16 - wait for initiator make present, then state 2
                %17 - wait for completor make present, then state 3

DLDEV - Device logical device number
  IOT  I/O TYPE  -  I/O System type
                      0 = Series II / III  I/O system
                      1 = HP-IB Systems
                      2 = unused
                      3 = unused

DSAVE - Device processing flags
  RD  READDONE    - A card has already been read.
  AF  ABORTFLAG   - A device clear has already been sent for
                      this series of aborted IOQs.
```

2608 Line Printer DIT (HP-IB Systems)

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element (however, there is only one device per 2608 controller.) The following diagram shows the DIT used for the 2608 line printer driver.

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
      +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
     0| 0| 0|AC|RQ| 0| 0| 0|IO|IA|NO|ST| 0|   STATE   |   DFLAG
      +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
     1| SYSDB relative pointer to the DIT for the next|   DLINK
      | device requesting this resource or service    |
      +-----------------------------------------------+
     2| IOQ table relative index to the first IOQ in  |   DIOQP
      | request list for this device                  |
      +--------+--------------+-----------------------+
     3|              Logical device number            |   DLDEV
      +--------+--------------+-----------------------+
     4| SYSDB relative pointer to Device Linkage Table|   DDLTP
      +-----------------------------------------------+
     5| SYSDB relative pntr to Interrupt Linkage Table|   DILTP
      +-----------------------------------------------+
     6|VM|      |    TAB    |              |PS|FL|TP|   DSAVE
      +-----------------------------------------------+
     7| Hardware error pointer. Set when the driver   |   DSERR
      | detects an error.  Whenever <>0, the driver   |
      | monitor logs an I/O error and clears this word|
      +--+-----+-----------+-----------------+--+--+--+
   %10| Bit 0 is set at completion of timer           |   DTIME
      +--+-----+-----------+-----------------+--+--+--+
   %11| Holds the time out request entry index while  |   DRQST
      | a timer is active.                            |
      +-----------------------------------------------+
   %12| IOT |///////////////////|      Physical Unit # |   DUNIT
      +-----------------------------------------------+
   %13|          Hardware logged error status         | DLOGERROR
      +-----------------------------------------------+
```

```
DFLAG - Flags and request state
  AC  ACTIVE  - A monitor is currently servicing this device.
  RQ  REQUEST - A service request is pending while the monitor is
                active.
  IO  IOPROG  - An I/O Channel Program is running for this device.
  IA  IAK     - An interrupt or response has occurred for this device.
  NO  NOTRDY  - Go to state %10 after Idle Channel Program is started.
  ST  STWAIT  - The device monitor is starting an Idle Channel Program
                for this device.  There is no IOQ associated with this
                type of request.
```

```
STATE      - State of the device monitor.  Specifies the next action
             to be taken in SIODM in servicing the request:
               0 - start new request
               1 - not used
               2 - call driver initiator procedure
               3 - call driver completor procedure
               4 - not used
               5 - process request completed
               6 - initiate device recognition sequence
               7 - start operator intervention wait
             %10 - wait for interrupt (operator intervention)
                   restart at state 0
             %11 - wait for data segment freeze, then state 2
             %12 - wait for driver initiator to be frozen, then
                   allocate controller (state 2)
             %13 - wait for I/O completion interrupt, then state 3
             %14 - wait for controller, then call driver initiator
             %15 - not used
             %16 - wait for initiator make present, then state 2
             %17 - wait for completor make present, then state 3

DLDEV - I/O system type, unit and logical device number
  IOT I/O TYPE- Type of I/O system
              0 - HP3000 Series II/III
              1 - HP3000 HP-IB Systems
              2 - unused
              3 - unused

DSAVE - Device processing flags
  VM  VFCMOD    - VFC has been modified.
  TAB TABDFAULT - System tab default.
  PS  PRESPACE  - Last request used prespacing.
  FL  FULL      - Line printer buffer is full.
  TP  TOP       - Printer is at top of form
```

2608 Line Printer Status

```
BYTE 1 & BYTE 2:
BITS       USE

  0    ON LINE

  1    NOT READY
  2    VFC CHANNEL 9 (BOTTOM OF FORM)
  3    VFC CHANNEL 12 (TOP OF FORM)

  4    VFC INITIALIZED
  5    6/8 LINES PER INCH
  6    (NOT USED)

  7    POWER RESTORED/UNIT RESET
  8    ON LINE
  9    PRINT MECH ERROR

 10    SELF TEST FAILURE
 11    PAPER ERROR
 12    SELF TEST MODE

 13    6/8 LPI
 14    PLATEN/RIBBON ERROR
 15    (NOT USED)

BYTE  3:  PRINT MODE
          BITS 0-7  MODE NUMBER
BYTE  4:  PRIMARY/SECONDARY
          BITS 0-3  SECONDARY CHARACTER SET CODE
          BITS 4-7  PRIMARY CHARACTER SET CODE
BYTE  5:  SELF TEST
          BITS 0    PASS FAIL
          BITS 1-7  SUBTEST NUMBER
BYTE  6:  6 LPI DOT ROW COUNT
BYTE  7:  6 LPI FORM LINE NUMBER
BYTE  8:  6 LPI FORM LENGTH IN LINES
BYTE  9:  8 LPI DOT ROW COUNT
BYTE 10:  8 LPI FORM LINE NUMBER
BYTE 11:  8 LPI FORM LENGTH IN LINES
BYTE 12:  FIRMWARE IDENTIFICATION CODE
BYTE 20:  POWER-UP LANGUAGE
          BITS 0-3  SECONDARY CHARACTER SET CODE
          BITS 4-7  PRIMARY CHARACTER SET CODE
```

HP 2619A or 2613 Line Printer DIT (HP-IB Systems)

There is one DIT per physical device. If a physical device represents more than one logical device, the logical device number is obtained from the IOQ element (however, there is only one device per 2631 controller.) The following diagram shows the DIT used for the 2631 line printer driver.

```
       0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
      +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
     0| 0| 0|AC|RQ| 0| 0| 0|IO|IA|NO|ST| 0|   STATE   |   DFLAG
      +--+--+--+--+--+--+--+--+--+--+--+--+-----------+
     1| SYSDB relative pointer to the DIT for the next|   DLINK
      | device requesting this resource or service    |
      +-----------------------------------------------+
     2| IOQ table relative index to the first IOQ in  |   DIOQP
      | request list for this device                  |
      +-----+----------------+------------------------+
     3|            Logical device number              |   DLDEV
      +-----+----------------+------------------------+
     4| SYSDB relative pointer to Device Linkage Table|   DDLTP
      +-----------------------------------------------+
     5| SYSDB relative pntr to Interrupt Linkage Table|   DILTP
      +-----------------------------------------------+
     6|                              |BJ|AB|PS|FL|TP|   DSAVE
      +-----------------------------------------------+
     7| Hardware error status.  Set when the driver   |   DSERR
      | detects an error.  Whenever <>0, the driver   |
      | monitor logs an I/O error and clears this word|
      +-----------------------------------------------+
   %10| Bit 0 is set at completion of timer           |   DTIME
      +-----------------------------------------------+
   %11| Holds the time out request entry index while  |   DRQST
      | a timer is active.                            |
      +-----------------------------------------------+
   %12| IOT |///////////////////|      Physical unit # |   DUNIT
      +-----------------------------------------------+
   %13|          Hardware logged error status         | DLOGERROR
      +-----------------------------------------------+
```

```
DFLAG - Flags and request state
  AC  ACTIVE  - A monitor is currently servicing this device.
  RQ  REQUEST - A service request is pending while the monitor is
                active.
  IO  IOPROG  - An I/O Channel Program is running for this device.
  IA  IAK     - An interrupt or response has occurred for this device.
  NO  NOTRDY  - Go to state %10 after Idle Channel Program is started.
  ST  STWAIT  - The device monitor is starting an Idle Channel Program
                for this device.  There is no IOQ associated with this
                type of request.
```

```
STATE      - State of the device monitor.  Specifies the next action
             to be taken in SIODM in servicing the request:
               0 - start new request
               1 - not used
               2 - call driver initiator procedure
               3 - call driver completor procedure
               4 - not used
               5 - process request completed
               6 - initiate device recognition sequence
               7 - start operator intervention wait
             %10 - wait for interrupt (operator intervention)
                   restart at state 0
             %11 - wait for data segment freeze, then state 2
             %12 - wait for driver initiator to be frozen, then
                   allocate controller (state 2)
             %13 - wait for I/O completion interrupt, then state 3
             %14 - wait for controller, then call driver initiator
             %15 - not used
             %16 - wait for initiator make present, then state 2
             %17 - wait for completor make present, then state 3

DLDEV - I/O system type, unit and logical device number
  IOT I/O TYPE - Type of I/O system
           0 - HP3000 Series 2/3
           1 - HP3000 HP-IB Systems
           2 - Unused
           3 - Unused

DSAVE - Device processing flags
  BJ   BETJOB    - Between jobs flag. If set, suppress
                   Powerfail message.
  AB   ABORT     - Abort (caused by Powerfail or Operator)
                   has occurred.
  PS   PRESPACE  - Last request used prespacing.
  FL   FULL      - Line printer buffer is full.
  TP   TOP       - Printer is at top of form
```

HP 2680A/2688A DIT

```
        0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
        +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  DIT0  !0 !0 !AC!RQ!0 !0 !SP!CP!IA!NR!SW!  !   STATE   !  DFLAG
        -------------------------------------------------
     1  !             POINTER TO NEXT DIT               !  DLINK
        -------------------------------------------------
     2  !           INDEX TO ACTIVE IOQ OR ZERO         !  DIOQP
        -------------------------------------------------
     3  !            LOGICAL DEVICE NUMBER              !  DLDEV
        -------------------------------------------------
     4  !        DRIVER LINKAGE TABLE POINTER           !  DDLTP
        -------------------------------------------------
     5  !        INTERRUPT LINKAGE TABLE POINTER        !  DILTP
        -------------------------------------------------
     6  !     SPECIAL ERROR CONDITIONS TO BE LOGGED     !  DSTAT
        -------------------------------------------------
     7  !            ERROR LOGGING INFORMATION          !  DSERR
        -------------------------------------------------
     8  !T !    TIMEOUT INDICATION IN BIT 0             !  DTIME
        -------------------------------------------------
     9  !       TIMER REQUEST INDEX (TRL) OR ZERO       !  DTRLX
        -------------------------------------------------
    10  ! IOT !//////////////////!    PHYSICAL UNIT #   !  DUNIT
        !-----------------------------------------------!
    11  !          CURRENT DATA WRITE BYTE COUNT        !  DCBCNT
        -------------------------------------------------
    12  !             CURRENT DATA WORD COUNT           !  DCWCNT
        -------------------------------------------------
    13  !           # OF WORDS LEFT TO TRANSFER         !  DRCNT
        -------------------------------------------------
    14  ! BUFFER OFFSET FOR NEXT # OF WORDS TO XFER.    !  DOFFSET
        -------------------------------------------------
    15  !                                            !D!  DDEBUG
        -------------------------------------------------
    16  ! I/O STATUS BLOCK WORD 1 GETS LOGGED FROM HERE !  DLOGBUFFER
        -------------------------------------------------
    17  ! I/O STATUS BLOCK WORD 3 GETS LOGGED FROM HERE !
        -------------------------------------------------
 18/33  ! I/O STATUS AREA (16 WORDS, SEE DEFINITION)    !  DIOSTAT
        +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

```
DFLAG - DEVICE RELATIVE FLAGS.
      AC         ACTIVE BIT. 1 IMPLIES A MONITOR CURRENTLY
                 SERVICING THIS DEVICE.
      RQ         REQUEST BIT. 1 IMPLIES SERVICE REQUESTED
                 WHILE MONITOR IS ACTIVE.
      SP         SIO PREEMPTION. IF SET THEN A PREEMPTIVE
                 REQUEST HAS BEEN QUEUED FOR THIS DEVICE.
                 PREEMPT CODE IS SET IN IOQ ELEMENT.
      CP         CHANNEL PROGRAM IN PROGRESS. IF SET, THEN
                 A CHANNEL PROGRAM IS CURRENTLY EXECUTING.
      IA         IF SET, AN INTERRUPT OR RESPONSE HAS OCCURRED.
```

```
      NR         IF SET, DEVICE IS IN A NOT READY OR OPERATOR WAIT.
      SW         IF SET, AN IDLE CHANNEL PROGRAM SHOULD BE STARTED
                 FOR THIS DEVICE.
      MSTATE     CURRENT DRIVER STATE AS DEFINED BY THE MONITOR.
                 ALLOWABLE STATES ARE:
                 0  - START REQUEST
                 1  - NOT USED(BUT RESERVED)
                 2  - CALL DRIVER INITIATOR
                 3  - CALL DRIVER COMPLETOR
                 4  - UNUSED(BUT RESERVED)
                 5  - COMPLETE REQUEST..PERHAPS RETURN TO USER.
                 6  - UNEXPECTED INTERRUPT OCCURRED.
                 7  - START OPERATOR INTERVENTION WAIT.
               %10  - WAITING (ON OPERATOR). RESTART AT 0.
                11  - WAITING (DATA MAKEPRESENT/FREEZING)
                12  - WAITING (INITIATOR CODE MAKEPRESENT/FREEZE)
                13  - WAITING (FOR COMPLETION INTERRUPT)
                14  - WAITING (FOR DEVICE CONTROLLER AVAILABILITY)
                15  - UNUSED(BUT RESERVED)
                16  - WAITING (INITIATOR CODE MAKEPRESENT)
                17  - WAITING (COMPLETOR CODE MAKEPRESENT)

DLDEV - I/O SYSTEM TYPE, UNIT AND LOGICAL DEVICE NUMBER.
      IOT        I/O SYSTEM TYPE.
                 0 - HP3000 SERIES II/III (SIO/DIO)
                 1 - HP-IB Systems
                 2 - RESERVED
                 3 - RESERVED

DCBCNT - CURRENT BYTE COUNT TO BE TRANSFERRED.

DCWCNT - CURRENT WORD COUNT TO BE TRANSFERRED.

DRCNT  - REMAINING WORD COUNT TO TRANSFER.

DOFFSET - OFFSET IN BUFFER OF NEXT # WORDS TO TRANSFER.

DDEBUG  - IF BIT 15=1 THEN DEBUGGING INFO WILL BE SENT TO CONSOLE

DLOGBUFFER - STATUS WORDS 1 & 3 ARE MOVED HERE TO BE LOGGED
             IF THEY WERE LOGGED FROM THE I/O STATUS BLOCK
             THEIR CONTENTS MIGHT BE CHANGED BEFORE THEY
             WERE LOGGED.

DIOSTAT - I/O STATUS AREA 16 WORDS, SEE I/O STATUS BLOCK DEFINITION.
```

I/O Status Block

```
       0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0  !0 !--THE "OR" OF WORDS 1/15 IS LOCATED HERE----! DIT 17
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   1  !OF!MS!PW!PE!TE!  !  !  !  !  !  !  !  !  !  !  !     18
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   2  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !     19
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   3  !                MCS FAULT NUMBER               !     20
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   4  !CL!FL!VL!CU!FU!VU!IL!IP!ST!SB!IR!MP!NJ!NM!TL!NC!     21
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   5  !LP!PF!NC!  !  !  (RESERVED)  !  !  !  !  !  !  !     22
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   6  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !     23
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   7  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !     24
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   8  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !     25
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   9  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !     26
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  10  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !     27
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  11  !  !  !  !  !  !  (RESERVED)  !  !  !  !  !  !  !     28
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  12  !            RECORD NUMBER OF ERROR             !     29
      +--               IF WORD 4 IS                --+
  13  !                  NON-ZERO                     !     30
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  14  !  SHEET NUMBER OF ERROR IF WORD 4 IS NON-ZERO  !     31
      +--                   OR                      --+
  15  !  LAST SHEET TRANSFERRED IF "JOB" & POWER-ON   !     32
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

```
WORD 0 - EACH BIT IS THE 'OR' OF ONE WORD IN THE TABLE (EXCEPT
         BIT 0 WHICH IS NOT USED).  THEREFORE, BIT .(1:1) IS SET
         IF WORD 1 IN THE TABLE IS NON-ZERO.

WORD 1 - BIT= 0 - (OF)  ONLINE/OFFLINE BIT.
              1 - (MS)  MESSAGE BEING DISPLAYED ON THE 2680A/2688A CONSOLE.
              2 - (PW)  POWER UP COMPLETED SINCE LAST I/O STATUS READ.
              3 - (PE)  PARITY ERROR DETECTED ON PHI COMMAND.
              4 - (TE)  TRANSMISSION ERROR DETECTED IN THE PRINTER.
           5/15 -       RESERVED.  UNUSED.

WORD 2 - NOT USED. RESERVED.

WORD 3 - MCS FAULT NUMBER.  CONTAINS AN INTEGER DESCRIBING THE LAST
         FAULT TO OCCUR SINCE THE LAST TIME THE I/O STATUS WAS READ
         OR THE HP 2680A/2688A WAS POWERED DOWN.  IF THE WORD IS ZERO THERE
```

IS NO MCS FAULT. SEE DCS ERS FOR A DESCRIPTION OF THE MCS FAULT NUMBERS.

```
WORD 4 - BIT= 0 - (CL)  NO ROOM FOR ATTEMPTED CHARACTER SET LOAD.
              1 - (FL)  NO ROOM FOR ATTEMPTED FORM LOAD.
              2 - (VL)  NO ROOM FOR ATTEMPTED VFC LOAD.
              3 - (CU)  ATTEMPT TO PRINT DATA AND THERE IS NO CURRENTLY
                  SELECTED CHARACTER SET.
              4 - (FU)  ATTEMPT TO SELECT AN UNDEFINED FORM SET.
              5 - (VU)  ATTEMPT TO PRINT DATA AND THERE IS NO CURRENTLY
                  SELECTED VFC SET.
              6 - (IL)  ATTEMPT TO PRINT DATA AND THERE IS NO CURRENTLY
                  SELECTED LOGICAL PAGE TABLE (LPT) ENTRY.
              7 - (IP)  ATTEMPT TO MOVE PEN OFF THE LOGICAL PAGE.
              8 - (ST)  THE 2680A/2688A COULD NOT PROCESS ALL OF THE DATA
                  BEFORE IT WAS SUPPOSED TO BE TRANSFERRED TO THE
                  DRUM/PAPER.  DATA WAS LOST!
              9 - (SB)  SPOOLER BLOCK CONTAINS FORMAT ERROR.
             10 - (IR)  INVALID RECOVERY BLOCK RECEIVED FROM SPOOLER.
             11 - (MP)  MAXIMUM NUMBER OF COPIES PER PHYSICAL PAGE
                  HAS BEEN EXCEEDED.  THIS IS A RESULT OF THE
                  SPOOLER PROCESS SETTING THE MAXIMUM COPIES PER
                  PAGE WITH FUNCTION CODE 132.
             12 - (NJ)  A COMMAND OR FUNCTION CODE WAS RECEIVED WHEN NO
                  "JOB" WAS IN PROGRESS.  THE COMMAND OR FUNCTION WAS
                  IGNORED BY THE DCS.
             13 - (NM)  NO MEMORY.  2680A/2688A DYNAMIC MEMORY ALLOCATION HAS
                  DETECTED THAT MAIN MEMORY IS COMPLETELY OCCUPIED WITH
                  CHARACTER SETS, VFC'S, FORMS AND DATA SUCH THAT THE
                  2680A/2688A CANNOT PROCESS THE CURRENT INPUT DATA.  DATA
                  WILL BE LOST!
             14 - (TL)  ATTEMPT TO PRINT DATA AND THERE ARE MORE THAN
                  THE MAXIMUM ALLOWABLE LOGICAL PAGE TABLE (LPT)
                  ENTRIES SELECTED.
             15 - (NC)  A NON-EXISTENT VFC CHANNEL WAS SKIPPED TO.

WORD 5 - BIT= 0 - (LP)  LOGICAL PAGE TRUNCATED TO FIT PHYSICAL PAGE.
              1 - (PF)  PAGE SIZE REQUIRED BY PROGRAMMER DID NOT
                  MATCH PAGE SIZE SET BY OPERATOR.  OPERATOR PAGE
                  SIZE PREVAILS.
              2 - (NC)  NO CHARACTER SET SELECTED.

WORDS 6/11        NOT USED BUT RESERVED FOR FUTURE USE.

WORDS 12/13 - THE RECORD NUMBER WHICH CONTAINS THE OFFENDING ERROR
              AS DEFINED BY WORD FOUR.  IF A POWER FAIL OCCURS DURING
              A "JOB", THE POWER FAIL BIT IS SET AND A SHEET NUMBER IS
              MADE AVAILABLE IN WORDS FOURTEEN AND FIFTEEN.  HOWEVER,
              THE RECORD NUMBER IS LOST AND CANNOT BE REPORTED.  THESE
              WORDS OCCUR IN A "JOB" ONLY.

WORDS 14/15 - THE SHEET NUMBER ON WHICH THE ERROR OCCURRED AS DEFINED
              BY WORD FOUR.  IF AN ERROR OCCURS IN THE ENVIRONMENT FILE
              AT THE START OF A "JOB", THEN THIS NUMBER WILL BE ZERO.
```

IN ADDITION, WHEN A POWER FAIL OCCURS DURING A "JOB", THE POWER ON BIT IS SET IN WORD ONE AND THE SHEET NUMBER OF THE LAST SUCCESSFULLY TRANSFERRED PAGE IS PLACED HERE. THIS INFORMATION IS FOR USE BY THE SPOOLER SHOULD A RECOVERY OF A "JOB" BE DETERMINED. THESE WORDS OCCUR IN "JOB" ONLY.

ALL WORDS OF THE I/O STATUS ARE CLEARED WHENEVER THE STATUS BLOCK IS RETURNED TO THE HOST. IT IS UP TO THE HOST CPU TO RETAIN ANY ONGOING STATUS BITS REQUIRED.

```
QMISC -

       0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
      +--+--+--+-----+--+--+--+--+--------+--------+--+
IOQ3  !MB!RB!AB!IO!TO!            !  XFER  ! PARITY !  !  QMISC
      +--+--+--+-----+--+--+--+--+--------+--------+--+

WHERE:

  .(0:1) - MB          USER REQUESTED TRANSFER IN EXCESS OF 4096
                       WORDS.  THE DRIVER CAN WRITE UP TO 4096 WORDS
                       TO THE 2680A/2688A.  IN ORDER TO HANDLE UP TO 32K
                       WORDS, MULTIPLE WRITES ARE USED WITHOUT A
                       RETURN TO THE USER WHO CALLED THE DRIVER.
                       THIS BIT INDICATES THAT MULTIPLE WRITES ARE
                       BEING DONE TO THE 2680A/2688A.

  .(1:1) - RB          THE CURRENT WRITE BLOCK MUST BE RETRIED.

  .(2:1) - AB          USER REQUESTED ABORT IN PROGRESS FLAG.

  .(3:1) - IO          I/O STATUS HAS BEEN READ AND IS AVAILABLE.

  .(4:1) - TO          GENERAL I/O CONTROLLER TIMED OUT.

  .(5:4) - RESERVED    NOT CURRENTLY USED.

  .(9:3) - XFER        2680A/2688A TRANSFER ERROR COUNTER.

  .(12:3)- PARITY      CHANNEL PROGRAM COMMAND PARITY ERROR COUNTER.

  .(15:1)- RESERVED    NOT CURRENTLY USED.

   **NOTE**  IN THE ABOVE, SINGLE BIT FIELDS ARE AS DEFINED
             WHEN THE BIT IS A LOGIC "1".
```

Disc Request Table and Disc Requests

Requests for disc transfers are effected by acquiring an entry from the Disc Request Table (DISCREQTAB), filling the proper information, and calling the DISCQMANAGER to link the request into the device's doubly linked request qu queue.
The head and tail of a device's request queue are contained in the devices' DIT.

DISCREQTAB

```
                   +------------+
                   |            |
                   |            |
                   |            |
            +----->+------------+
+------+    |      |  Device's  |
|      |----+      |  Current   |
| DIT  |-------+   |  Request   |
|      |----+  |   +------------+
+------+    |  |   |            |
            |  |   |            |
            |  |   |            |
            |  |   |            |
            |  |   |            |
            |  |   |            |
            |  |   +------------+
            |  +-->|   First    |
            |      |  Request   |
            |      |  in Queue  |<>----+
            |      +------------+      |
            |      |            |      |
            |      |            |      |
            |      |            |      |
            |      |            |      |
            |      |------------|      |
            |      |   Next     |<>----+
            |      |  Request   |
            |      |  in Queue  |<>----+
            |      |------------|      |
            |      |            |      |
            |      +------------+      |
            |      |   Last     |<>----+
            +----->|  Request   |
                   |  in Queue  |
                   +------------+
                   |            |
                   |            |
                   |            |
                   +------------+
```

Disc Request Table

DISCREQTAB DST ENTRY# = 56 (%70)
DISCREQTAB PRT = %1017

Disc Request Table Entry 0 Format

```
               0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
              |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
DISCREQTAB00  |                 TOTAL ENTRIES                  |
              |------------------------------------------------|
DISCREQTAB01  |                ENTRY SIZE (%21)                |
              |------------------------------------------------|
DISCREQTAB02  |                PRIMARY ENTRIES                 |
              |------------------------------------------------|
DISCREQTAB03  |              IMPEDED PROCESS PCB               |
              |------------------------------------------------|
DISCREQTAB04  | TABLE INDEX OF HEAD OF AVAILABLE ENTRY LIST    |
              |------------------------------------------------|
DISCREQTAB05  | TABLE INDEX OF TAIL OF AVAILABLE ENTRY LIST    |
              |------------------------------------------------|
DISCREQTAB06  |               MAX ENTRIES IN USE               |
              |------------------------------------------------|
DISCREQTAB07  |             CURRENT ENTRIES IN USE             |
              |------------------------------------------------|
DISCREQTAB08  |                   OVERFLOWS                    |
              |------------------------------------------------|
DISCREQTAB09  |                                                |
              |                 TOTAL REQUESTS                 |
DISCREQTAB10  |                                                |
              |------------------------------------------------|
DISCREQTAB11  |SYSBASE INDEX OF HEAD OF DISABLED REQ Q         |  DISCQHEAD
              |------------------------------------------------|
DISCREQTAB12  |SYSBASE INDEX OF TAIL OF DISABLED REQ Q         |  DISCQTAIL
              |------------------------------------------------|
DISCREQTAB13  |   SERIAL WRITE QUEUE  HEAD                     |  SERWQHEAD
              |------------------------------------------------|
DISCREQTAB14  |A |/////////////////|MAX. SERIAL WRITE QUEUE    |   A = Active
              |------------------------------------------------|
DISCREQTAB15  |////////////////////////////////////////////////|
              |------------------------------------------------|
DISCREQTAB16  |////////////////////////////////////////////////|
              |------------------------------------------------|
```

Disc Request Element Format

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
Word 00  |A |M |D |S |I |B |C |D |M |Q |S |P |C |D |L |I |
         |B |M |I |B |O |K |O |A |M |U |I |F |U |I |D |N |
         |O |R |A |U |W |D |M |T |E |E |O |A |R |S |R |L |
         |R |E |G |F |A |  |P |A |R |U |F |I |R |A |  |O |
         |T |Q |  |  |K |  |  |F |R |E |A |L |E |B |  |C |
         |  |  |  |  |E |  |  |  |  |  |I |  |Q |  |  |A |
         |  |  |  |  |  |  |  |  |  |  |L |  |  |  |  |L |
         |-----------------------------------------------|
Word 01  |          REQUEST URGENCY CLASS                | URGCLASS
         |-----------------------------------------------|
Word 02  |          LOGICAL DEVICE NUMBER                | LDEVN
         |-----------------------------------------------|
Word 03  |              MISCELLANEOUS                    | MISC
         |-----------------------------------------------|
Word 04  | S|         DST       (IF PROCESS DISC I/O)    | DSTN
         |- - - - - - - - - - - - - - - - - - - - - - - -| S=STACK
         |          BANK      (IF SEGMENT TRANSFER)      |
         |-----------------------------------------------|
Word 05  | OFFSET INTO DATA SEG  (IF PROCESS DISC I/O)   | ADDR
         |- - - - - - - - - - - - - - - - - - - - - - - -|
         |     ADDRESS IN BANK   (IF SEGMENT TRANSFER)   |
         |-----------------------------------------------|
Word 06  |      UNIT #          |       FUNCTION         | FUNC
         |-----------------------------------------------|
Word 07  |        COUNT/XLOG/CONTROL RETURNS             | XFERCNT
         |-----------------------------------------------|
Word 08  |        P1 (HODA IF SEGMENT TRANSFER           | PAR1
         |-----------------------------------------------|
Word 09  |        P2 (LODA IF SEGMENT TRANSFER           | PAR2
         |-----------------------------------------------|
Word 10  |///////////////////////////| QUALIFIER | STATUS | STAT
         |-----------------------------------------------|
Word 11  |FREE|           PCB NUMBER                     | PCBN
         |-----------------------------------------------|
Word 12  |       INDEX OF PREV REQUEST IN QUEUE          | PREVREQP
         |-----------------------------------------------|
Word 13  |       INDEX OF NEXT REQUEST IN QUEUE          | NEXTREQP
         |-----------------------------------------------|
Word 14  |                                               |
         |-       SEGIDENTIFIER  (IF SEG TRANSFER)      -|SEGIDENT
Word 15  |                                               |
         |-----------------------------------------------|
Word 16  |DISPLACEMENT OF READ OR WRITE FROM SEG BASE(MM)|SEGDISP
         |-----------------------------------------------|
```

Note: Upon return to free list, word (#1) becomes index of next EE free entry.

Word 0 - QFLAG - Request dependent flags

Bit 0	.ABORT	Request has been aborted externally.
Bit 1	.MMREQ	Request is for a segment transfer.
Bit 2	.DIAG	Diagnostic request (not used).
Bit 3	.SBUF	System Buffer. Target is a system buffer whose index is relative to the start of the SBUF table.
Bit 4	.IOWAKE	Wake caller on completion of request.
Bit 5	.BLOCKED	Blocked I/O. Caller is waited in ATTACHIO until request is completed.
Bit 6	.COMPLETED	Request has been completed and caller woken if he had specified.
Bit 7	.DATAFRZN	Data segment has been made present and is frozen.
Bit 8	.MAMERRORD	MAM error on data segment make present.
Bit 9	.PREQQUEUED	Request is queued into disc's req queue
Bit 10	.SFAIL	Start SIO failure in GIP.
Bit 11	.PFAIL	The I/O has been aborted because of a powerfail.
Bit 12	.CURREQ	Request is device's current request.
Bit 13	.DISABLED	Request is disabled.
Bit 14	.LDR	Request in local DRQ.
Bit 15	.INLOCAL	Buffer DST is in process locality.

Word 2 - QLDEV.QLDEVN - Logical Device Number
Word 3 - QMISC - Device dependent.

Word 4
QDSTN - If SYSBUFRs is clear then this is the DST number of the target data segment. If bit 0 is set then buffer address is a DB offset value instead of segment relative offset (implemented for NOWAIT IO and NOBUFF).
Word 5
QADDR - Offset in data segment or sys buff table to target data buffer.
Word 6
QFUNC.FUNC - Function code and qualifiers as specified by driver.

Word 7
QXFERCNT-On initiation specifies the word count if positive or byte count if negative. At completion of the request this location contains the actual transmission count in the same units as the call. Certain control requests return data through this location.
Word 8
QPAR1 - Parameter one, defined by driver
Word 9
QPAR2 - Parameter two, defined by driver
QMISC - Miscellaneous request dependent storage available to driver.
Word 10
QSTAT.PCBN - PCB Number of process which made this request. Zero if not associated with any process and IOQ is to be returned by the system.
.QUALIFIER - A code which further defies or qualifies the general status. Defined by driver.
.STATUS - General Status. Indicates current and result state of the request according to the following codes.
0 - not started or awaiting completion.
1 - successful completion.
2 - end of file detected.
3 - unusual condition.
4 - irrecoverable error.

NOTE: See I/O System Status Returns.

Word 11 - bit 0=1 Q element is on free list.

IOQ Table Layout

```
                  |-----------------------|
             0    |        TOTAL #        |
                  |-----------------------|
             1    |      ENTRY SIZE       |
                  |-----------------------|
             2    |       PRIMARY #       |
                  |-----------------------|
             3    |  IMPEDED PROCESS PCB  |
                  |-----------------------|
          ----4-- |      HEAD INDEX       |   THEAD
          |       |-----------------------|
 ----------5--    |      TAIL INDEX       |   TTAIL
 |        |       |-----------------------|
 |        |  6    |  MAXIMUM OF IN USE    |   TUSE
 |        |       |-----------------------|
 |        |  7    |   CURRENT IN USE      |
 |        |       |-----------------------|
 |        |  10   |      OVERFLOWS        |   TOVRFL
 |        |       |-----------------------|
 |        |  11   |                       |
 |        |       |    TOTAL REQUESTS     |   TRQSTS
 |        |  12   |                       |
 |        |       |-----------------------|
 |        |  13   | UNUSED                |
 |        |       |-----------------------|
 |        |    -->|                       |
 |        |    |  |-----------------------|
 |        |    |  |      INDEX OF 5       |-----
 |        |    |  |-----------------------|    |
 |        |    |  |                       |    |
 |        |    |  |       ENTRY 1         |    |
 |        |    |  |                       |    |
 |        |    |  |-----------------------|    |
 |-----|---|---|->|                       |<---|----
          |    |  |-----------------------|    |   |
          |    |  |           0           |    |   |
          |    |  |-----------------------|    |   |
          |    |  |                       |    |   |
          |    |  |        ENTRY 2        |    |   |
          |    |  |                       |    |   |
          |    |  |-----------------------|    |   |
          |---|-->|                       |    |   |
               |  |-----------------------|    |   |
               |--|      INDEX OF 1       |    |   |
                  |-----------------------|    |   |
```

IOQ (Cont.)

```
|                        |     |    |
|                        |     |    |
|        ENTRY 3         |     |    |
|                        |     |    |
|                        |     |    |
|------------------------|     |    |
|                        |     |    |
|------------------------|     |    |
|     Indeterminate      |     |    |
|------------------------|     |    |
|                        |     |    |
|                        |     |    |
|        ENTRY 4         |     |    |
|       (IN USE)         |     |    |
|                        |     |    |
|                        |     |    |
|------------------------|     |    |
|                        |<----|    |
|------------------------|          |
|      INDEX OF 2        |----------|
|------------------------|
|                        |
|                        |
|        ENTRY 5         |
|                        |
|                        |
|------------------------|
```

I/O Queue Element (IOQ)

```
   0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  |      REQUEST DEPENDENT FLAGS                  |
 0|                                               |  QFLAG
  |-----------------------------------------------|
 1|                 IOQ POINTER                   |  QLINK
  |-----------------------------------------------|
 2|                   QLDEVN                      |  QLDEV
  |-----------------------------------------------|
 3|                MISCELLANEOUS                  |  QMISC
  |-----------------------------------------------|
 4|S |        DATA SEGMENT DST NUMBER             |  QDSTN S(Word 4(0:1)
  |                                               |  Stackflag  If set
  |-----------------------------------------------|  QADDR is DB rel.
  |                                               |
 5|                   ADDRESS                     |  QADDR
  |-----------------------------------------------|
 6|      UNIT            |         FUNCTION       |  QFUNC
  |-----------------------------------------------|
 7|        COUNT/XLOG/CONTROL RETURNS             |  QWBCT
  |-----------------------------------------------|
 8|                     P1                        |  QPAR1
  |-----------------------------------------------|
 9|                     P2                        |  QPAR2
  |-----------------------------------------------|
10|//////////////////////|  QUALIFIER    | STATUS |  QSTAT
  |-----------------------------------------------|
11|FR|                  PCBN                      |  QPCBN
  |-----------------------------------------------|
```

QFLAG - Request dependent flags

Bit 0 .ABORT Request has been aborted externally.

Bit 1 .SPECIAL Special handling is to be applied to this request. For disc, indicates a memory management request.

Bit 2 .DIAG Diagnostic request (not used).

Bit 3 .SBUF System Buffer. Target is a system buffer whose index is relative to the start of the SBUF table.

Bit 4 .IOWAKE Wake caller on completion of request.

Bit 5 .BLOCKED Blocked I/O. Caller is waited in ATTACHIO until request is completed.

Bit 6 .COMPLETED Request has been completed and caller woken if he had specified.

I/O Queue Element (Cont.)

Bit 7 .DATAFRZN Data segment has been made present and is frozen.

Bit 8 .MAMERRORD MAM error on data segment make present.

Bit 9 .PREQ This request has been started but was preempted by a MAM request.

Bit 10 .SFAIL Start SIO failure in GIP.

Bit 11 .PFAIL The I/O has been aborted because of a powerfail.

Bits12-13 .PREEMPT Preemptive type code: 1-soft, 2-hard.

Bit 15 .MSGDONE A message request reply has completed.

QLINK - Table relative index of next IOQ element. Points to first word of element.

QLDEV - Logical Device Number

QMISC - Device dependent.

QDSTN - If SYSBUFRs is clear then this is the DST number of the target data segment. If bit 0 is set then buffer address is a DB offset value instead of segment relative offset (implemented for NOWAIT IO and NOBUFF).

QADDR - Offset in data segment or sys buff table to target data buffer.

QFUNC.FUNC - Function code and qualifiers as specified by driver.

QWBCT - On initiation specifies the word count if positive or byte count if negative. At completion of the request this location contains the actual transmission count in the same units as the call. Certain control requests return data through this location.

QPAR1 - Parameter one, defined by driver

QPAR2 - Parameter two, defined by driver

QMISC - Miscellaneous request dependent storage available to driver.

QPCBN - PCB Number of process which made this request. Zero if not associated with any process and IOQ is to be returned by the system.

.QUALIFIER - A code which further defies or qualifies the general status. Defined by driver.

.STATUS - General Status. Indicates current and result state of the request according to the following codes.

0 - not started or awaiting completion.
1 - successful completion.
2 - end of file detected.
3 - unusual condition.
4 - irrecoverable error.

Word 11 bit 0- Queue element is on free list.

I/O System Status Returns

	STATUS %
0 - PENDING	
1 - WAITING FOR COMPLETION	10
2 - DOING ERROR RECOVERY	20
3 - NOT READY WAIT	30
4 - NO WRITE RING WAIT	40
5 - NEW PAPER TAPE WAIT	50
1 - SUCCESSFUL	
0 - NORMAL	1
1 - READ TERMINATED WITH SPECIAL CHARACTER	11
2 - TAPE RETRY FOR SUCCESS REQUIRED	21
3 - LOW TAPE OR END OF TAPE AFTER WRITE	31
2 - END OF FILE	
1 - PHYSICAL END OF FILE	12
2 - DATA	22
3 - END OF DATA	32
4 - HELLO	42
5 - BYE	52
6 - JOB	62
7 - END OF JOB	72
3 - UNUSUAL CONDITION	
1 - TERMINAL PARITY ERROR	13
2 - TERMINAL READ TIMED OUT	23
3 - I/O ABORTED EXTERNALLY	33
4 - DATA LOST	43
5 - DATA SET NOT READY OR DISCONNECT OR UNIT NOT ON LINE	53
6 - ABORTED BECAUSE OF POWER FAIL	63
7 - BOT AND BSR, BSF REQUEST	73
10 - TAPE RUNAWAY	103
11 - EOT AND WRITE REQUEST	113
12 - NO WRITE RING AFTER REQUEST TO OPERATOR	123
13 - END OF TAPE (PAPER TAPE LOW)	133
14 - PLOTTER LIMIT SWITCH REACHED	143
15 - ENABLE SUBSYSTEM BREAK AND NO CONTROL Y PIN	153
16 - READ TIME RETURNED OVERFLOW	163
17 - BREAK STOPPED READ	173
20 - WRITE AND NO CARD IN WAIT STATION	203
21 - DEVICE POWERED ON - OPERATING ENVIRONMENT LOST	213
27 - VFC HAS BEEN RESET	273

I/O System Status Returns (Cont.)

4 - IRRECOVERABLE ERROR

0 - INVALID REQUEST	4
1 - TRANSMISSION ERROR	14
2 - I/O TIME OUT	24
3 - TIMING ERROR	34
4 - SIO FAILURE	44
5 - UNIT FAILURE	54
6 - INVALID DISC ADDRESS	64
7 - TAPE PARITY ERROR	74
11 - PAPER TAPE TAPE ERROR	114
12 - SYSTEM ERROR	124
13 - INVALID SBUF INDEX	134
14 - CHANNEL FAILURE, TIMEOUT OR NO RESPONSE FROM CONTROLLER	144
15 - UNINITIALIZED MEDIA (LINUS)	154
16 - NO SPARE BLOCKS AVAILABLE	164
17 - DELETED RECORD DETECTED ON IBM FLOPPY DISC	174
20 - LABELED DEVICE UNAVAILABLE AFTER REELSWITCH	204
21 - PARITY ERROR DETECTED ON PHI COMMAND (EPOC)	214

5 - ERROR IN DATA CONTROL INFORMATION

		XLOG
0 - INVALID ITEM NUMBER	5	
1 - INVALID ACCESS FOR ITEM	15	VALID ACCESS
2 - FAILURE IN FOPEN OR FREAD	25	FS ERROR NUMBER
3 - PARITY CHANGE IN 8 BIT MODE	35	
4 - INVALID INFO. FILE FORMAT	45	
5 - CHECKSUM ERROR IN INFO FILE	55	
6 - PASSED VALUE LESS THAN MIN.	65	MIN.VALUE ALLOWED
7 - PASSED VALUE GREATER THAN MAX.	75	MAX.VALUE ALLOWED
10 - PASSED VALUE IS UNSUPPORTED	105	
11 - COUNT LESS THAN REQUIRED TO RETURN ALL INFO.	115	MIN.SPACE NEEDED
12 - COUNT GREATER THAN AVAILABLE TO STORE INFO.	125	MAX.SPACE AVAIL
13 - PASSED VALUES NOT IN ASCENDING ORDER	135	OFFSET OF ELEMENT
14 - PASSED CHARACTER HAS OTHER DEFINED FUNCTION	145	OTHER FUNCTION

I/O Queue Element for 7976A Magnetic Tape

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0|      Request dependent flags (see below)      |   QFLAG
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   1| SYSDB relative pointer to next IOQ element.   |   QLINK
    | Points to first word of element.              |
    +-----------------------+-----------------------+
   2|               logical device number           |   QLDEV
    +--+--+--+--+-----+-----+--------+--------+-----+
   3| R| B| F| G|BO| TOUT| FSCNTR | BSCNTR | RTCNTR |   QMISC
    +--+--+--+--+-----+--------+--------+--------+--+
   4| S| If QFLAG.(3:1) is clear then this is the   |   QDSTN
    |  | DST number of the target data segment.  If |
    |  | S is set, QADDR is DB relative.            |
    +--+--------------------------------------------+
   5| Offset in the data segment or system buffer   |   QADDR
    | table to the target data buffer.              |
    +-----------------------+-----------------------+
   6|                       | Function code for     |   QFUNC
    |                       | this request.  (See   |
    |                       | next section.)        |
    +-----------------------+-----------------------+
   7| On initiation, specifies the word count (>0)  |   QWBCT
    | or byte count (<0).  At completion of the     |
    | request this location contains the actual     |
    | transmission count in the same units (bytes   |
    | or words) as in the request.                  |
    +-----------------------------------------------+
 %10| Parameter 1.  Used only for reads.  Contains  |   QPAR1
    | the EOF specification in bits (13:3).         |
    +-----------------------------------------------+
 %11| Parameter 2.  Used only for writes.  If bit   |   QPAR2
    | (13:1) is set, writing past EOT is allowed.   |
    +-----------------------+-------------+---------+
 %12|///////////////////////|  QUALIFIER  | STATUS  |   QSTAT
    +-----------------------+-------------+---------+
 %13|                   PCB NUMBER                  |
    |-----------------------------------------------|
```

QFLAG - Request dependent flags

Bit 0 ABORT - Abort this request and return an error indication to the caller.

Bit 1 SPECIAL - Apply special handling to this request. (Not used)
Bit 2 DIAG - This is a request from the diagnostic subsystem. (Not used)
Bit 3 SYSBUFF - Target is an index relative to the SBUF Table of the data buffer.
Bit 4 IOWAKE - Wake caller on completion of request.
Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTACHIO

until the request is completed. Implies IOWAKE.

Bit 6 COMPLETED - The request has been completed and the caller awakened if he had requested (with IOWAKE).
Bit 7 DATAFRZN - Set by the memory management routines (MAM) when a MAKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.
Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.
Bit 9 PREQ - (Not used)
Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to START'HPIB resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.
Bit 11 PFAIL - The request was aborted because of a system power failure.

QMISC - Driver request dependent flags and counters. Used mostly for error retries.

RETRY - Indicates an error retry is in progress.
BACK - Backspace record processing for an error retry is in progress.
FORWARD - Forward space record processing for an error retry is in progress.
GAP - Gap processing for an error retry is in progress.
BODEOF - Backspace record due to a data EOF processing is in progress.
TOUTCNTR - GIC timed-out counter.
FSCNTR - Forward space record counter.
BSCNTR - Backspace record counter.
RTCNTR - Error retry counter.

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.
STATUS - General status indicating the final state of the request. The following codes are used:
0 - Not started or awaiting completion.
1 - Successful completion.
2 - End-of-file detected.
3 - Unusual, but recoverable, condition detected.
4 - Irrecoverable error has occurred.
QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

I/O Queue Element (IOQ) for CIPER

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0|      Request dependent flags (see below)      |   QFLAG
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   1| IOQ table index to the next IOQ element.      |   QLINK
    | Points to first word of element.              |
    +-----------------------+-----------------------+
   2|                        Logical device number  |   QLDEV
    +-----------------------+-----------------------+
   3|                                               |   QMISC
    +--+--------------------------------------------+
   4|  | If QFLAG.(3:1) is clear then this is the   |   QDSTN
    | S| DST number of the target data segment.  If |
    |  | S is set, QADDR is DB relative.            |
    +--+--------------------------------------------+
   5| Offset in the data segment or system buffer   |   QADDR
    | table to the target data buffer.              |
    +-----------------------+-----------------------+
   6|                       | Function code for     |   QFUNC
    |                       | this request.  (See   |
    |                       | next section.)        |
    +-----------------------+-----------------------+
   7| On initiation, specifies the word count (>0)  |   QWBCT
    | or byte count (<0).  At completion of the     |
    | request this location contains the actual     |
    | transmission count in the same units (bytes   |
    | or words) as in the request.                  |
    +-----------------------------------------------+
 %10| Parameter 1.                                  |   QPAR1
    +-----------------------------------------------+
 %11| Parameter 2.                                  |   QPAR2
    +-----------------------+-------------+---------+
 %12|                       |  QUALIFIER  |RSTATUS  |   QSTAT
    +-----------------------+-------------+---------+
 %13|                                          PCBN |   QPCB
    +-----------------------------------------------+
```

QFLAG - Request dependent flags

Bit 0 ABORT - Abort this request and return an error indication to the caller.
Bit 1 SPECIAL - Apply special handling to this request. (Not used)
Bit 2 DIAG - This is a request from the diagnostic subsystem.
Bit 3 SYSBUFF - Target is an index relative to the SBUF Table of the data buffer.
Bit 4 IOWAKE - Wake caller on completion of request.
Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies IOWAKE.
Bit 6 COMPLETED - The request has been completed and the caller awakened if he had requested (with IOWAKE).
Bit 7 DATAFRZN - Set by the memory management routines (MAM) when a

MAKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.

Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.

Bit 9 PREQ - (Not used)

Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 PFAIL - The request was aborted because of a system power failure.

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.

RSTATUS - General status indicating the final state of the request. The following codes are used:

0 - Not started or awaiting completion.
1 - Successful completion.
2 - End-of-file detected.
3 - Unusual, but recoverable, condition detected.
4 - Irrecoverable error has occurred.

QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

HP-IB CIPER Physical Driver Request Codes

OPERATION	FUNCTION	PARAMETERS
READ	0	None
WRITE	1	None
FILE OPEN	2	None
FILE CLOSE	3	None
DEVICE CLOSE	4	None
CIPER INIT	184	None

CIPER Driver Return Status Codes

General Status (13:3)	Qualifying Status (8:5)	Overall (8:8)
0 - Pending	1 - Waiting For Completion	%10
	3 - Not Ready Wait	%30
1 - Successful	0 - No Errors	%1
2 - End of File	(Not Used)	
3 - Unusual Condition	3 - Request Aborted	%33
	6 - Powerfail Abort	%63
	%21 - Device Powered Up	%213
4 - Irrecoverable Error	0 - Invalid Request	%4
	1 - Transfer Error	%14
	2 - I/O Timed Out Before Complete	%24
	4 - SIO Failure	%44
	5 - Unit Failure	%54
	%12 - System Error	%124
	%14 - Channel Failure	%144
	%21 - Parity Error	%214

2608 Line Printer I/O Queue Element (HP-IB Systems)

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0|      Request dependent flags (see below)      |   QFLAG
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   1| SYSDB relative pointer to next IOQ element.   |   QLINK
    | Points to first word of element.              |
    +-----------------------+-----------------------+
   2|              Logical device number            |   QLDEV
    +--+--+--+--------+-----+-----------+-----------+
   3|PP|PE|MC|TOUTCNTR|                 | WAITCODE  |   QMISC
    +--+--+--+--------+-----------------+-----------+
   4| S| If QFLAG.(3:1) is clear then this is the   |   QDSTN
    |  | DST number of the target data segment.  If |
    |  | S is set, QADDR is DB relative.            |
    +--+--------------------------------------------+
   5| Offset in the data segment or system buffer   |   QADDR
    | table to the target data buffer.              |
    +-----------------------+-----------------------+
   6|                       | Function code for     |   QFUNC
    |                       | this request.  (See   |
    |                       | next section.)        |
    +-----------------------+-----------------------+
   7| On initiation, specifies the word count (>0)  |   QWBCT
    | or byte count (<0).  At completion of the     |
    | request this location contains the actual     |
    | transmission count in the same units (bytes   |
    | or words) as in the request.                  |
    +-----------------------------------------------+
 %10| Parameter 1.  Vertical Format specification.  |   QPAR1
    | (See next section for detail.)                |
```

```
    +-----------------------------------------------+
 %11| Parameter 2.  Space Mode Flags.  (See next    |   QPAR2
    | section for details.)                         |
    +-----------------------+---------------+-------+
 %12|///////////////////////|  QUALIFIER    |STATUS |   QSTAT
    +-----------------------+---------------+-------+
 %13|                  PCB NUMBER                   |   QPCBN
    |-----------------------------------------------|
```

QFLAG - Request dependent flags

Bit 0 ABORT - Abort this request and return an error indication to the caller.

Bit 1 SPECIAL - Apply special handling to this request. (Not used)

Bit 2 DIAG - This is a request from the diagnostic subsystem. (Not used)

Bit 3 SYSBUFF - Target is an index relative to the SBUF Table of the data buffer.

Bit 4 IOWAKE - Wake caller on completion of request.

Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTACHIO until the request is completed. Implies IOWAKE.

Bit 6 COMPLETED - The request has been completed and the caller awakened if he had requested (with IOWAKE).

Bit 7 DATAFRZN - Set by the memory management routines (MAM) when a MAKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.

Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.

Bit 9 PREQ - (Not used)

Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.

Bit 11 PFAIL - The request was aborted because of a system power failure.

QMISC - Driver request dependent flags and counters.

PRE'TO'POST - Pre to post spacing change flag.

PEJECT - Last operation was a page eject.

MASTERCLR - Master clear done to clear powerfail bit in status. Master clear needs to be done from not ready condition.

TOUTCNTR - Channel time-out retry counter.

WAITCODE - Indicates type of wait:
0 - new request
1 - completion wait
2 - not ready wait

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.

STATUS - General status indicating the final state of the request. The following codes are used:
0 - Not started or awaiting completion.
1 - Successful completion.
2 - End-of-file detected.
3 - Unusual, but recoverable, condition detected.
4 - Irrecoverable error has occurred.

QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

2608 Line Printer Request Codes

Operation	Function	Parameters
WRITE	1	P1 - Vertical Format Specification 1 - use 1st data char as format spec %53 - "+", print and suppress spacing %55 - "-", print and triple space %60 - "0", print and double space %61 - "1", print and top of form %200-%277, print and space N-%200 lines %300-%377, print with channel N-%277 All others, print and single space. P2 - Space Mode Flags (15:1) - Prespace flag if set, print then fill buffer if clear, fill buffer then print (14:1) - No page stepover flag if set, single and double space without stepover (66 lines/page) if clear, single and double space with stepover (60 lines/page)
FILE OPEN	2	Page eject if not at top of form
FILE CLOSE	3	Page eject if not at top of form
DEVICE CLOSE	4	Page eject if not at top of form
READ STATUS	%17	Read I/O status Count - buffer must be at least 2 bytes
VFC SET	%100	Load VFC RAM Count - form length in words (0 loads RAM form internal ROM) P1 - 6 for 6 LPI or 8 for 8 LPI any other value defaults to 6 LPI
TAB SET	%101	Sets logical column definition P1 - 0 to 15, any other value defaults to 15

2619A & 2631 Line Printer IOQ Element (HP-IB Systems)

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   0|      Request dependent flags (see below)      |   QFLAG
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
   1| SYSDB relative pointer to next IOQ element.   |   QLINK
    | Points to first word of element.              |
    +-----------------------+-----------------------+
   2|             Logical device number             |   QLDEV
    +--+--+--+--------+-----+-----------+-----------+
   3|PP|PE|PF|TOUTCNTR|                 | WAITCODE  |   QMISC
    +--+--+--+--------+-----------------+-----------+
   4| S| If QFLAG.(3:1) is clear then this is the   |   QDSTN
    |  | DST number of the target data segment.  If |
    |  | S is set, QADDR is DB relative.            |
    +--+--------------------------------------------+
   5| Offset in the data segment or system buffer   |   QADDR
    | table to the target data buffer.              |
    +-----------------------+-----------------------+
   6|                       | Function code for     |   QFUNC
    |                       | this request.  (See   |
    |                       | next section.)        |
    +-----------------------+-----------------------+
   7| On initiation, specifies the word count (>0)  |   QWBCT
    | or byte count (<0).  At completion of the     |
    | request this location contains the actual     |
    | transmission count in the same units (bytes   |
    | or words) as in the request.                  |
    +-----------------------------------------------+
 %10| Parameter 1.  Vertical Format specification.  |   QPAR1
    | (See next section for detail.)                |
    +-----------------------------------------------+
 %11| Parameter 2.  Space Mode Flags.  (See next    |   QPAR2
    | section for details.)                         |
    +-----------------------+-------------+---------+
 %12|///////////////////////|  QUALIFIER  | STATUS  |   QSTAT
    +-----------------------+-------------+---------+
 %13|                  PCB NUMBER                   |   QPCBN
    |-----------------------------------------------|
```

QFLAG - Request dependent flags

Bit 0 ABORT - Abort this request and return an error indication to the caller.
Bit 1 SPECIAL - Apply special handling to this request. (Not used)
Bit 2 DIAG - This is a request from the diagnostic subsystem. (Not used)
Bit 3 SYSBUFF - Target is an index relative to the SBUF Table of the data buffer.
Bit 4 IOWAKE - Wake caller on completion of request.
Bit 5 BLOCKED - Blocked I/O. The caller is waited in ATTACHIO

until the request is completed. Implies IOWAKE.
Bit 6 COMPLETED - The request has been completed and the caller awakened if he had requested (with IOWAKE).

Bit 7 DATAFRZN - Set by the memory management routines (MAM) when a MAKEPRESENT request is successfully completed and indicates the data segment is frozen in memory.
Bit 8 MAMERRORD - An error has occurred while MAM was trying to make the target data segment present and freeze it in memory.
Bit 9 PREQ - (Not used)
Bit 10 SFAIL - Delayed failure of SIO instruction. If a call to STARTIO resulted in the request being added to the channel queue, this bit indicates that the SIO instruction failed when the request was selected for execution.
Bit 11 PFAIL - The request was aborted because of a system power failure.

QMISC - Driver request dependent flags and counters for 2631.

PRE'TO'POST - Pre to post spacing change flag.
PEJECT - Last operation was a page eject.
TOUTCNTR - Channel time-out retry counter.
POWERFAIL - Power fail flag indicates power fail occurred.
WAITCODE - Indicates type of wait:
0 - new request
1 - completion wait
2 - not ready wait

Format for 2619A

```
 0  1  2  3  4                          12        15
----------------------------------------------------
|PP|PE|PF|TO|BF|                        | WAITCODE  |
----------------------------------------------------
```

TOUT - Channel timed out flag
BUF'FILL - Buffer fill operation in progress

QSTAT - PCB number and request completion status.

PCBN - The Process Control Block (PCB) number of the process which made this request. If zero, the request is not associated with any process and the IOQ element is to be returned by the system when the request has completed.
STATUS - General status indicating the final state of the request. The following codes are used:
0 - Not started or awaiting completion.
1 - Successful completion.
2 - End-of-file detected.
3 - Unusual, but recoverable, condition detected.
4 - Irrecoverable error has occurred.
QUALIFIER - A code which further defines or qualifies the general status. (See the section Driver Return Status Codes.)

2619 Line Printer Request Codes

```
Operation    Function          Parameters

WRITE          1          P1 - Vertical Format Specification
                             1 - Use 1st data char as format
                                 specification.

                             %53 - "+", print and suppress spacing
                             %55 - "-", print and triple space
                             %60 - "0", print and double space
                             %61 - "1", print and top of form

                            %200-%277, print and space N-%200 lines
                            %300-%312, print with channel N-%277

                            %320 - Fill Line Printer Buffer Only

                            All others, print and single space.

                          P2 - Space Mode Flags
                               (15:1) - Prespace flag
                                  if set, print then fill buffer
                                  if clear, fill buffer then print
                               (14:1) - No page stepover flag
                                  if set, single and double space
                                    without stepover (66 lines/page)
                                  if clear, single and double space
                                    with stepover (60 lines/page)

FILE OPEN      2          Page eject if not at top of form

FILE CLOSE     3          Page eject if not at top of form

DEVICE CLOSE   4          Page eject if not at top of form

READ STATUS    %17        Read I/O status
                            Count - buffer size

*IDENTIFY      %110       Return ID value in Bank & Buffaddr

*SELF TEST:
  INITIATE     %111       Subtest number to execute in Bank and Buffaddr
                            (subtest number ranges from 0 to 7)
  STATUS       %112       Subtest result returned in Bank & Buffaddr

*LOOPBACK TEST:
  WRT DATA     %113       Data to LP in Bank & Buffaddr [PING]
  READ DATA    %114       Data from LP read into Bank & Buffaddr [PONG]
                          Count - Buffer Size (256 bytes max)
```

2631 Line Printer Request Codes (HP-IB)

```
Operation    Function          Parameters

WRITE          1          P1 - Vertical Format Specification
                             1 - Use 1st data char as format
                                 specification.

                             %53 - "+", print and suppress spacing
                             %55 - "-", print and triple space
                             %60 - "0", print and double space
                             %61 - "1", print and top of form

                            %200-%277, print and space N-%200 lines
                            %300-%307, print with channel N-%277

                            %320 - Fill Line Printer Buffer Only

                            All others, print and single space.

                          P2 - Space Mode Flags
                               (15:1) - Prespace flag
                                  if set, print then fill buffer
                                  if clear, fill buffer then print
                               (14:1) - No page stepover flag
                                  if set, single and double space
                                    without stepover (66 lines/page)
                                  if clear, single and double space
                                    with stepover (60 lines/page)

FILE OPEN      2          Page eject if not at top of form

FILE CLOSE     3          Page eject if not at top of form

DEVICE CLOSE   4          Page eject if not at top of form

READ STATUS    %17        Read I/O status
                            Count - 1 byte minimum required

VFC SET        %100       LOADS VFC RAM
                           P1 -  1 - 1 LPI (lines per inch)
                                 2 - 2 LPI
                                 3 - 3 LPI
                                 4 - 4 LPI
                                 5 - 5 LPI
                                 6 - 6 LPI
                                 8 - 8 LPI
                                12 - 12 LPI
                                Any other value defaults to 6 LPI.
```

I/O Queue Element For HP-IB Card Reader

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  0|      Request dependent flags (see below)       |   QFLAG
   +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  1| SYSDB relative pointer to next IOQ element.    |   QLINK
   | Points to first word of element.               |
   +-----------------------+------------------------+
  2|             Logical device number              |   QLDEV
   +--+--+--+--+--+-----+--+------+--------+--------+
  3|       Auxiliary buffer flag.                   |   QMISC
   +--+--+--+--+--+-----+---------+--------+--------+
  4| S| If QFLAG.(3:1) is clear then this is the    |   QDSTN
   |  | DST number of the target data segment.  If  |
   |  | S is set, QADDR is DB relative.             |
   +--+---------------------------------------------+
  5| Offset in the data segment or system buffer    |   QADDR
   | table to the target data buffer.               |
   +-----------------------+------------------------+
  6|                       | Function code for      |   QFUNC
   |                       | this request.  (See    |
   |                       | next section.)         |
   +-----------------------+------------------------+
  7| On initiation, specifies the word count (>0)   |   QWBCT
   | or byte count (<0).  At completion of the      |
   | request this location contains the actual      |
   | transmission count in the same units (bytes    |
   | or words) as in the request.                   |
   +------------------------------------------------+
%10| Parameter 1.  Contains the EOF specification   |   QPAR1
   +------------------------------------------------+
%11| Parameter 2.  Contains the data mode           |   QPAR2
   | specification in bits (11:2). (See below card  |
   | reader request codes for detail information)   |
   +-----------------------+---------------+--------+
%12|///////////////////////|  QUALIFIER    | STATUS |   QSTAT
   +-----------------------+---------------+--------+
%13|                   PCB NUMBER                   |   QPCBN
   |------------------------------------------------|
```

QFLAG - Request dependent flags

```
  Bit 0  ABORT     - Abort this request and return an error indication
                     to the caller.
  Bit 1  SPECIAL   - Apply special handling to this request.  (Not used)
  Bit 2  DIAG      - This is a request from the diagnostic subsystem.
  Bit 3  SYSBUFF   - Target is an index relative to the SBUF Table of
                     the data buffer.
  Bit 4  IOWAKE    - Wake caller on completion of request.
  Bit 5  BLOCKED   - Blocked I/O.  The caller is waited in ATTACHIO
                     until the request is completed.  Implies IOWAKE.
```

```
  Bit 6  COMPLETED - The request has been completed and the caller
                     awakened if he had requested (with IOWAKE).
  Bit 7  DATAFRZN  - Set by the memory management routines (MAM) when a
                     MAKEPRESENT request is successfully completed and
                     indicates the data segment is frozen in memory.
  Bit 8  MAMERRORD - An error has occurred while MAM was trying to
                     make the target data segment present and freeze
                     it in memory.
  Bit 9  PREQ      - (Not used)
  Bit 10 SFAIL     - Delayed failure of SIO instruction.  If a call to
                     STARTIO resulted in the request being added to
                     the channel queue, this bit indicates that the SIO
                     instruction failed when the request was selected
                     for execution.
  Bit 11 PFAIL     - The request was aborted because of a system power
                     failure.
```

QMISC - Auxiliary buffer flag used to indicated a read into the driver's buffer and not the user's buffer.

QSTAT - PCB number and request completion status.

```
  PCBN      - The Process Control Block (PCB) number of the process
              which made this request.  If zero, the request is not
              associated with any process and the IOQ element is to
              be returned by the system when the request has completed.
  STATUS    - General status indicating the final state of the request.
              The following codes are used:
                0 - Not started or awaiting completion.
                1 - Successful completion.
                2 - End-of-file detected.
                3 - Unusual, but recoverable, condition detected.
                4 - Irrecoverable error has occurred.
  QUALIFIER - A code which further defines or qualifies the general
                status.  (See the section Driver Return Status Codes.)
```

CS 80 Disc Request Queue Element (IOQ)

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
    +-----------------------------------------------+
   0|      Request dependent flags (see below)      |   QFLAG
    +-----------------------------------------------+
   1|      Request urgency class                    | QURGCLASS
    +-----------------------+-----------------------+
   2|                Logical device number          |   QLDEV
    +-----+--+--+--+--+-----+--+--+-----+-----------+
   3|CHANF|RS|OP|IM|SR|RTRAN|LF|SP|     | WAITCODE  |   QMISC
    +-----+--+--+--+--+-----+--+--+-----+-----------+
   4| S|   DST   (If process disc I/O)               |   QDSCTN
    |- - - - - - - - - - - - - - - - - - - - - - - -|
    |      DST   (If segment transfer) [S=Stack]    |
    +--+--------------------------------------------+
   5| Offset in the data seg (If process disc I/O)  |   QADDR
    |- - - - - - - - - - - - - - - - - - - - - - - -|
    | Address in Bank (If segment transfer)         |
    +-----------------------+-----------------------+
   6|        Unit #         | Function code for     |   QFUNC
    |                       | this request.         |
    +-----------------------+-----------------------+
   7| On initiation, specifies the word count (>0)  |   QWBCT
    | or byte count (<0).  At completion of the     |
    | request this location contains the actual     |
    | transmission count in the same units (bytes   |
    | or words) as in the request.                  |
    +-----------------------------------------------+
 %10| P1 - Parameter 1 (Usually High Order of       |   QPAR1
    | Current Logical Disc Address [CLDA1])         |
    +-----------------------------------------------+
 %11| P2 - Parameter 2 (Usually Low Order of        |   QPAR2
    | Current Logical Disc Address [CLDA2])         |
    +-----------------------+--------------+--------+
 %12|////////////////////// |  QUALIFIER   | STATUS |   QSTAT
    +-----------------------+--------------+--------+
 %13|     PCB                                       |
    +-----------------------------------------------+
 %14| Sysbase relative indx of previous req in queue| QPREVREQP
    +-----------------------------------------------+
 %15| Sysbase relative indx of next req in queue    | QNEXTREQP
    +-----------------------------------------------+
 %16|        Segidentifier (If segment transfer     | QSEGIDENT
    +-----------------------------------------------+
 %17| Displacement of read or wrt from seg base (MM)| QSEGDISP
    +-----------------------------------------------+
 %20|S |////////////////////////////////////////////|
    |W |////////////////////////////////////////////|
    |A |////////////////////////////////////////////|
    |P |////////////////////////////////////////////|
    +-----------------------------------------------+
```

QFLAG - Request dependent flags

```
  Bit 0  ABORT     - Request has been aborted externally.
  Bit 1  MMREQ     - Request is for a segment transfer.
  Bit 2  DIAG      - This is a request from the diagnostic subsystem.
  Bit 3  SBUF      - Target is an index relative to the SBUF Table of
                     the data buffer.
  Bit 4  IOWAKE    - Wake caller on completion of request.
  Bit 5  BLOCKED   - Blocked I/O.  The caller is waited in ATTACHIO
                     until the request is completed.  Implies IOWAKE.
  Bit 6  COMPLETED - The request has been completed and the caller
                     awakened if he had requested (with IOWAKE).
  Bit 7  DATAFRZN  - Data segment has been present and is frozen.
  Bit 8  MAMERRORD - An error has occurred while MAM was trying to
                     make the target data segment present and freeze
                     it in memory.
  Bit 9  PREQUEUED - Request is queued into disc's request queue
  Bit 10 SFAIL     - Delayed failure of SIO instruction.  If a call
                     to STARTIO resulted in the request being added
                     to the channel queue, this bit indicates that
                     the SIO instruction failed when the request was
                     selected for execution.
  Bit 11 PFAIL     - The request was aborted because of a system
                     power failure.
  Bit 12 CURREQ    - Request is device's current request.
  Bit 13 DISABLED  - Request is disabled.
  Bit 14 DISATMPT  - Attempt to disable this request.
  Bit 15 MSGDONE   - A message request reply has completed.
```

QLDEV.QLDEVN - Logical Device Number

QMISC - Driver request dependent flags and counters.

```
  CHAN'ERR'FLG   - Channel error retry flag.
  RSTAT'FAIL'FLG - Request status failed flag.
  OPER'REQ'FLG   - Operator requested release flag.
  IM'FAULT'FLG   - Internal maintenance fault flag.
  STAT'RTRY'FLG  - Status error single retry flag.
  RTRANS'FLG     - Retransmit required flag.
  LOAD'FLG       - Media load flag.
  SYS'PFAIL'FLG  - System powerfail flag.

  WAITCODE       - Indicates type of wait:

                   0 - new request
                   1 - completion wait
                   2 - not ready wait
                   3 - release/release deny wait
                   4 - IOQ defer wait
                   5 - DSCT read wait
                   6 - DSCT write wait
                   7 - synchronization wait
```

```
QDSTN - If system buffer is clear then this is the DST
        number of the target data segment. If bit 0 is
        set then buffer address is a DB offset value
        instead of segment relative offset (implemented
        for NOWAIT I/O and NOBUFF).

QADDR - Offset in data segment or system buffer table to
        target data buffer.

QFUNC - Function code and qualifiers as specified by
        driver.

QSTAT - PCB number and request completion status.

  PCBN     - The Process Control Block (PCB) number of the process
             which made this request.  If zero, the request is not
             associated with any process and the IOQ element is to
             be returned by the system when the request has completed.

  STATUS   - General status indicating the final state of the request.

             0 - Not started or awaiting completion.
             1 - Successful completion.
             2 - End-of-file detected.
             3 - Unusual, but recoverable, condition detected.
             4 - Irrecoverable error has occurred.

  QUALIFIER - A code which further defines or qualifies the general
              status.  (See the section Driver Return Status Codes.)
```

CS 80 Integrated Cartridge Tape Request

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
    +-----------------------------------------------+
   0|      Request dependent flags (see below)      |   QFLAG
    +-----------------------------------------------+
   1|      Request urgency class                    | QURGCLASS
    +-----------------------+-----------------------+
   2|                Logical device number          |   QLDEV
    +-----+--+--+--+--------+--+--+-----+-----------+
   3|CHANF|RS|OP|IM| RETRY  |LF|SP|     | WAITCODE  |   QMISC
    +-----+--+--+--+--------+--+--+-----+-----------+
   4| S|   DST   (If process disc I/O)               |   QDSCTN
    |- - - - - - - - - - - - - - - - - - - - - - - -|
    |      DST   (If segment transfer) [S=Stack]    |
    +--+--------------------------------------------+
   5| Offset in the data seg (If process disc I/O)  |   QADDR
    |- - - - - - - - - - - - - - - - - - - - - - - -|
    | Address in Bank (If segment transfer)         |
    +-----------------------+-----------------------+
   6|        Unit #         | Function code for     |   QFUNC
    |                       | this request.         |
    +-----------------------+-----------------------+
   7| On initiation, specifies the word count (>0)  |   QWBCT
    | or byte count (<0).  At completion of the     |
    | request this location contains the actual     |
    | transmission count in the same units (bytes   |
    | or words) as in the request.                  |
    +-----------------------------------------------+
 %10| P1 - Parameter 1 (Usually High Order of       |   QPAR1
    | Current Logical Disc Address [CLDA1])         |
    +-----------------------------------------------+
 %11| P2 - Parameter 2 (Usually Low Order of        |   QPAR2
    | Current Logical Disc Address [CLDA2])         |
    +-----------------------+--------------+--------+
 %12|         PCBN          |  QUALIFIER   | STATUS |   QSTAT
    +-----------------------+--------------+--------+
 %13| Sysbase relative indx of previous req in queue| QPREVREQP
    +-----------------------------------------------+
 %14| Sysbase relative indx of next req in queue    | QNEXTREQP
    +-----------------------------------------------+
 %15|        Segidentifier (If segment transfer     | QSEGIDENT
    +-----------------------------------------------+
 %16| Displacement of read or wrt from seg base (MM)| QSEGDISP
    +-----------------------------------------------+
 %17|S |////////////////////////////////////////////|
    |W |////////////////////////////////////////////|
    |A |////////////////////////////////////////////|
    |P |////////////////////////////////////////////|
    +-----------------------------------------------+
```

QFLAG - Request dependent flags

```
Bit 0  ABORT     - Request has been aborted externally.
Bit 1  MMREQ     - Request is for a segment transfer.
Bit 2  DIAG      - This is a request from the diagnostic subsystem.
Bit 3  SBUF      - Target is an index relative to the SBUF Table of
                   the data buffer.
Bit 4  IOWAKE    - Wake caller on completion of request.
Bit 5  BLOCKED   - Blocked I/O.  The caller is waited in ATTACHIO
                   until the request is completed.  Implies IOWAKE.
Bit 6  COMPLETED - The request has been completed and the caller
                   awakened if he had requested (with IOWAKE).
Bit 7  DATAFRZN  - Data segment has been present and is frozen.
Bit 8  MAMERRORD - An error has occurred while MAM was trying to
                   make the target data segment present and freeze
                   it in memory.
Bit 9  PREQUEUED - Request is queued into disc's request queue
Bit 10 SFAIL     - Delayed failure of SIO instruction.  If a call
                   to STARTIO resulted in the request being added
                   to the channel queue, this bit indicates that
                   the SIO instruction failed when the request was
                   selected for execution.
Bit 11 PFAIL     - The request was aborted because of a system
                   power failure.
Bit 12 CURREQ    - Request is device's current request.
Bit 13 DISABLED  - Request is disabled.
Bit 14 DISATMPT  - Attempt to disable this request.
Bit 15 MSGDONE   - A message request reply has completed.
```

QLDEV.QLDEVN - Logical Device Number

QMISC - Driver request dependent flags and counters.

```
  CHAN'ERR'FLG    - Channel error retry flag.
  RSTAT'FAIL'FLG  - Request status failed flag.
  OPER'REQ'FLG    - Operator requested release flag.
  IM'FAULT'FLG    - Internal maintenance fault flag.
  RETRY'COUNT     - Retry count area.
  LOAD'FLG        - Media load flag.
  SYS'PFAIL'FLG   - System powerfail flag.

  WAITCODE        - Indicates type of wait:

                    0 - new request
                    1 - completion wait
                    2 - not ready wait
                    3 - release/release deny wait
                    4 - IOQ defer wait
                    5 - DSCT read wait
                    6 - DSCT write wait
                    7 - synchronization wait
```

QDSTN - If system buffer is clear then this is the DST number of the target data segment. If bit 0 is set then buffer address is a DB offset value

instead of segment relative offset (implemented for NOWAIT I/O and NOBUFF).

QADDR - Offset in data segment or system buffer table to target data buffer.

QFUNC - Function code and qualifiers as specified by driver.

QSTAT - PCB number and request completion status.

```
  PCBN     - The Process Control Block (PCB) number of the process
             which made this request.  If zero, the request is not
             associated with any process and the IOQ element is to
             be returned by the system when the request has completed.

  STATUS  - General status indicating the final state of the request.

             0 - Not started or awaiting completion.
             1 - Successful completion.
             2 - End-of-file detected.
             3 - Unusual, but recoverable, condition detected.
             4 - Irrecoverable error has occurred.

  QUALIFIER - A code which further defines or qualifies the general
              status.  (See the section Driver Return Status Codes.)
```

SBUF Table Layout

```
         |--------------------------------------------|
         |                 TOTAL #                    |
         |--------------------------------------------|
         |               ENTRY SIZE                   |     TSIZE
         |--------------------------------------------|
         |                PRIMARY #                   |
         |--------------------------------------------|
         |           IMPEDED PROCESS PCB              |
         |--------------------------------------------|
    -----|               HEAD INDEX                   |     THEAD
    |    |--------------------------------------------|
    | ---|               TAIL INDEX                   |     TTAIL
    | |  |--------------------------------------------|
    | |  |           MAXIMUM OF IN USE                |     TUSE
    | |  |--------------------------------------------|
    | |  |             CURRENT IN USE                 |
    | |  |--------------------------------------------|
    | |  |               OVERFLOWS                    |     TOVRFL
    | |  |--------------------------------------------|
    | |  |                 TOTAL                      |
    | |  |                REQUESTS                    |     TRQSTS
    | |  |--------------------------------------------|
    | |  |               INDEX OF 5                   |---------
    | |  |--------------------------------------------|        |
    | | ->                                            |        |
    | | ||                ENTRY 1                     |        |
    | | ||                                            |        |
    | | ||--------------------------------------------|        |
    | | ||                   0                        |        |
    | | ||--------------------------------------------|        |
    | -->|                                            |<---    |
    |   ||                ENTRY 2                     |   |    |
    |   ||                                            |   |    |
    |   ||--------------------------------------------|   |    |
    |   -|               INDEX OF 1                   |   |    |
    |    |--------------------------------------------|   |    |
    ---->|                                            |   |    |
         |                ENTRY 3                     |   |    |
         |                                            |   |    |
         |--------------------------------------------|   |    |
         |               INDEX OF 2                   |----    |
         |--------------------------------------------|        |
         |                                            |        |
         |                ENTRY 4                     |<---    |
         |               (IN USE )                    |   |    |
         |--------------------------------------------|   |    |
         |               INDEX OF 4                   |----    |
         |--------------------------------------------|        |
         |                                            |<--------
         |                ENTRY 5                     |
         |                                            |
         |--------------------------------------------|

3 - 1 - 5 - 4 - 2
```

Table Element Allocation (SBUF)

The allocation of the elements in the IOQ terminal buffer (TBUF) and system buffer (SBUF) tables is of concern to the I/O system.

FREE LIST OF TABLE ELEMENTS

These tables are in the form of a free-linked list of the free elements. For the SBUF's the -1 word of entry is the link to the next element. For the TBUF's, word zero is the link and word 1 is the link for the IOQ elements.

Each word has an 11-word header beginning at the base of the table . The first six words of the header are for managing the table and the second five are for monitoring table activity.

The entries follow the header at word eleven.

ELEMENT ALLOCATION

Elements are obtained from the beginning of the free list, pointed to by the head and returned to the end of the free list pointed by the tail.

When the free list is empty, the head index is zero and the tail index is set to point at the head index.

The tables are divided into two areas: a primary and a secondary area. Most requests are obtained from the primary area. The secondary area is used only for critical requirements when the primary area is exhausted. These areas are logical areas determined by parameters in the header.

The utility of the core resident tables is seriously reduced if their use is not restricted to dynamic situations.

One of three responses must be specified to the routines which allocate elements from the I/O system tables.

1. Impede caller if primary is empty.

2. Get from primary area only.

3. Get from secondary area if primary area is empty.

Table Element Allocation (Cont.)

Request types 2 and 3 return an indication to the caller if the request could not be satisfied. The following table specifies the types of calls for element allocation and the action if an element is not activated.

BUFFER USER	CALL TYPE	FINAL ACTION
SBUF's		
File systen	Inpede	---
Ptape	Inpede	---
Bad track	Prinary	Forget request
IOQ's		
ATTACHIO (not inpedable)	Prinary	Return IOQX-0
ATTACHIO (inpedable)	Inpede	---
SIODM (menory nanagenent)	Secondary	Sudden death
IOMESSAGE	Secondary	I/O error

HEADER DEFINITION

```
Prinary #      - Nunber of elenents in the prinary area.
Total #        - Total nunber of elenents in the table.
Size           - Size in words of each elenent.
Inpeded PCB    - If not zero then contains the PCB nunber of the
                 first process waiting for an elenent in this table.
Head index     - Index of first free elenent.
Tail index     - Index of last free elenent.
In use         - Current nunber not in free list.
Overflows      - Nunber of requests nade for an elenent.
Total requests - Total nunber of elenents requested.
```

ICS Global

```
QI -
      |--------------------|
   63.                     .
     .  RESERVED           .
   50.                     .
      |--------------------|
   49|  CANDPIN            |
      |--------------------|
   48|  LAST WEIGHT        |
      |--------------------|
   47|                     |
      |  PAUSETIME         |
   46|                     |
      |--------------------|
   45|  LISTSTATE          |
      |--------------------|
   44|  CUREFILTER         |
      |--------------------|
   43|  CURDFILTER         |
      |--------------------|
   42|  CWTNUM             |
      |--------------------|
   41|  CWTDENOM           |
      |--------------------|
   40|  CURCFILTER         |
      |--------------------|
   39|  MAXCFILTER         |
      |--------------------|
   38|  MINCFILTER         |
      |--------------------|
   37|  ESCHEDBASE         |
      |--------------------|
   36|  DSCHEDBASE         |
      |--------------------|
   35|  CSCHEDBASE         |
      |--------------------|
   34|  WORSTEPRI          |
      |--------------------|
   33|  WORSTDPRI          |
      |--------------------|
   32|  WORSTCPRI          |
      |--------------------|
   31| MISC. BOUNDS FLAGS  |
      |--------------------|
   30| SYSTEM MEM BOUND    |
      |--------------------|
   29| XDS UPPER BOUND     |
      |--------------------|
   28| DL INITIAL          |
      |--------------------|
```

```
       |--------------------|
    27|                     |
       |--------------------|
    26|  XDS SEGMENT BANK   |   Series 64 only
       |--------------------|
    25|  XDS SEGMENT BASE   |   Series 64 only
       |--------------------|
    24|  XDS SEGMENT LIMIT  |   Series 64 only
       |--------------------|
    23|  PRIV BNDS STAT WD  |   Series 64 only
       |--------------------|
    22|                     |
      .                     .
      . RESERVED            .
      .                     .
    19|                     |
       |--------------------|
    18|       DISAP         |   PSEN, PSDB counter
       |--------------------|
    17|      Reserved       |
       |--------------------|
    16|       SDST          |   process' stack DST#
       |--------------------|
    15|       PSTA          |   pseudo-interrupt status
       |--------------------|
    14|       PADDR         |   pseudo-interrupt address
       |--------------------|
    13|     TRACE FLAG      |   flag set non-zero on IXIT away fron ICS
       |--------------------|
    12|       PFAIL         |   PTR to powerfail PCB
       |--------------------|
    11|       JCUT          |   absolute JCUT address
       |--------------------|
    10|        XP           |   pointer to executing process PCB
       |--------------------|
     9|       PCBX          |   absolute stack address
       |--------------------|
     8|        Z            |   stack DB relative Z
       |--------------------|
     7|        DL           |   stack DB relative DL
       |--------------------|
     6|        S            |   stack DB relative S
       |--------------------|
     5|       SBANK         |   stack bank
       |--------------------|
     4|       STDB          |   absolute stack DB
       |--------------------|\
     3|        0            ||
       |--------------------||
     2|        P            ||
       |--------------------| >  DISPATCH stack narker
     1|      STATUS         ||
       |--------------------||
QI   0|P |      0           ||
```

```
      |--------------------|/
   +1|   DB BANK RETURN    |\
      |--------------------| > FOR DISPATCH
      |      DB RETURN     ||
      |--------------------|/
      |D |      PARM       |
      |--------------------|
```

P=PSEUDO-DISABLED AND DISP INSTRUCTION EXECUTED.
D=DISPATCHER INTERRUPTED.

ICS Global Cells With Initial Values

STDB - absolute address of the currently running process's stack.
SBANK - bank address for process' stack.
S - stack DB relativeS
DL - stack DB relative DL
Z - stack DB relative Z
PCBX - absolute stack address
XP - PCB table relative pointer to word 0 of the running process' PCB.

The above cells are to be initialized for the PROGENITOR.

CPCB - absolute 4, is an absolute version of XP. If CPCB is zero, then the above cells are invalid. This will never be the case in a process. CPCB should also be set by INITIAL.
SDST - DST# for running process' stack.
JCUT - the bank zero absolute address of the JCUT table.
PADDR - PB relative address for the procedure PSEUDOINT.
PSTA - status value for PSEUDOINT, %140000+CST#.
DISAP - PSDB counter, initially 0.

INITIAL sets the above as described.

CS 80 Disc Interrupt Linkage Table (ILT)

There is one ILT for each device controller configured on the system. A controller may support more than one unit, however the CS'80 disc driver will only concern itself with the single unit controller.

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15   MNEMONIC
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    0|      Channel                                  |   ICPVA0
    1|          Program                              |   ICPVA1
    2|              Variable                         |   ICPVA2
    3|                   Area (ICPVA)                |   ICPVA3
     +-----------------------------------------------+
    4|      DMA Abort                                |   ICPVA4
    5|          Address                              |   ICPVA5
     +-----------------------------------------------+
    6|                    0                          |   ISRQL
     +--+--------------+-----+----------+------------+
    7|LI|    CHANQUE    |     |   CHAN   |    DEV     |   ICNTRL
     +--+--------------+-----+----------+------------+
  %10| SYSDB relative pointer to channel program area|   ISIOP
     +-----------------------------------------------+
  %11| SYSDB relative pointer to idle status area    |   ISTAP
     +-----------------------------------------------+
  %12| single instruction that is executed to extract|   IUNIT
     | the device unit number from the status pointed|
     | to by ISTAP. [Since only Unit 0 exists on the |
     | CS'80 discs, ANDI 0 is used to return Unit 0] |
     +-----------------------------------------------+
  %13| SYSDB relative DIT pointer of the device      |   ICDP
     | currently using the channel to perform a      |
     | data operation.                               |
     +-----------------------+-----------------------+
  %14|       SIOPSIZE        |        CQUEN          |   IQUEUE
     +--+--+--+--------------+-----------+-----------+
  %15|RW|WP|IG|                          |  HCUNIT   |   IFLAG
     +--+--+--+--------------------------+-----------+
  %16| SYSDB relative DIT pointer for unit 0         |   IDITP0
     +-----------------------------------------------+
  %17| 20 bytes status area for idle channel program |   ISTAT
     +-----------------------------------------------+
    .|                    .                          |
     +-----------------------------------------------+
    .|                    .                          |
     +-----------------------------------------------+
  %31|              CS'80 Discs                      |
    .                 Channel                        .
     |                Program                        |
     +-----------------------------------------------+
```

ICPVA0 - Channel Program Variable Area

The first word is used by the channel program processor to store status information after I/O channel aborts. The next word is used by the driver to indicate if status should be examined for special conditions or errors. The other two words are not used.

ICPVA4 - DMA abort address

If a DMA abort occurs, the absolute address where the abort occurred is stored in this area.

ICNTRL - Contains controller information

LIM -If this bit is set, the controller is sharing a software channel resource in order to limit bandwidth.

CHANQUE -The software channel resource number.

CHAN -Channel number (four most significant bits of DRTN).

DEV -Device number (three least significant bits of DRTN).

IQUEUE - The channel program contains:

SIOPSIZE - (number of words + 1)/2 in the channel program area.

CQUEN - or a multi-unit controller this field contains the software controller resource number.

IFLAG - Controller and Channel Program state flags

RUNWAIT - An Idle Channel Program should be started when there are no active requests to process.

WAITPROG - An Idle Channel Program has been started for this controller. This bit is reset by an interrupt.

IGNOREHI - An HIOP instruction has been issued against this controller but the channel program was not in a wait statement. Therefore ignore the interrupt generated by

HCUNIT the channel code when this program halts. - Highest configured unit number for this controller.

ISTAT - 20 bytes of status from the idle channel program.

CHAPTER 14 SPOOLING

Input Device Directory/Output Device Directory

IDD/ODD (Common attributes referred to as XDD)

```
       IDD:  DST = 45 (= %55)        ODD:  DST = 46 (= %56)
             SIR =  3                      SIR =  4
```

Overview of Table Structure

```
       +------------------------------------+
     0 |  Entry 0 (8 words)                 |
     . | . . . . . . . . . . . . . . . . .  |
     2 |  Subentry area pointer             | ---
     . | . . . . . . . . . . . . . . . . .  |    |
     7 |                                    |    |
       +------------------------------------+    |
       |                                    |    |
       |  Head entries (4 words each)       |    |
       |                                    |    |
       +------------------------------------+    |
       |                                    | <--
       |  Subentries (%40 words each)       |
       |                                    |
       +------------------------------------+
```

Entry 0 (Overall Table Definitions)

```
   0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
  +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
 0|       Maximum size     |       Current size   |0 (sectors)
  +------------------------+----------------------+
 1|  Head entry size = 4   | Subentry size = %40  |1 ( words )
  +------------------------+----------------------+
 2|   Subentry area pointer (segment relative)    |2
  +--+--------------------------------------------+
 3|DD|       Next avail device file ID (DFID)     |3
  +--+--------------------------------+-----------+
 4|///////////////////////////////////|   Fence   |4
  +-----------------------------------+-----------+
 5|///////////////////////////////////////////////|5
  +-----------------------------------------------+
 6|///////////////////////////////////////////////|6
  +-----------------------------------------------+
 7|///////////////////////////////////////////////|7
  +-----------------------------------------------+
```

DD: 0 ==> This is the IDD,
        1 ==> This is the ODD.

Fence: For spooled output devices (ODD), the system-wide outfence. For spooled input devices (IDD), the jobfence.

Typical Head Entry (4 words)

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
 |    Device outfence     |///////////////////////|
 +------------------------+-----------------------+
 |                  Head pointer                  |
 +------------------------------------------------+
 |                  Tail pointer                  |
 +------------------------------------------------+
 |                 Logical device                 |
 +------------------------------------------------+
```

There are two types of head entry, a class entry and a logical device entry. There is only one class entry, if it exists at all, and it is the first head entry in the XDD. All spoolfiles opened by class (e.g., LP, SLOWLP, EPOC, PP, etc.) are linked to this entry. There is one logical device entry for each real (physical, as opposed to virtual) device on the system. Output devices appear in the ODD, input devices in the IDD. AC/DC devices such as terminals appear in both directories.
Each head entry is linked to 0 or more subentries (a typical subentry is shown in the next table). A null chain (0 subentries) consists of head pointer = 0 and tail pointer = segment-relative address of the associated head pointer. If one or more subentries exists, the pointers are segment-relative addresses of the first word of the first and last subentries of the chain. Any intermediate subentries are linked through the subentries. The tail subentry always contains a 0-link.
The Device Outfence and LDEV# fields are meaningless for the class entry. For logical device entries (non-0 Logical Device field), a non-0 Device Outfence means that this outfence overrides the system-wide outfence in word 4 of entry 0, but only for this device.

Typical Subentry (%40 words)

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     +--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--+
  %0|//|State|  Outpri   |CL|/////////////////////////|0
     +--+--+--+-----------+--+-------------------------+
  %1| Type|               Job number                   |1
     +-----+-------------------------------------------+
  %2|                                                  |2
  %3|                   User name                      |3
  %4|                                                  |4
  %5|                                                  |5
     +-------------------------------------------------+
  %6|                                                  |6
  %7|                 Account name                     |7
 %10|                                                  |8
 %11|                                                  |9
     +-------------------------------------------------+
 %12|                                                  |10
 %13|                   Job name                       |11
 %14|                                                  |12
 %15|                                                  |13
     +-------------------------------------------------+
 %16|                                                  |14
 %17|                   File name                      |15
 %20|                                                  |16
 %21|                                                  |17
     +--+----------------------------------------------+
 %22|ID|              Device file ID                   |18
     +--+--+--+----------------------------------------+
 %23|FS|DA|//|   XDD head index (see explanation)      |19
     +--+--+--+----------------------------------------+
 %24|  Logical device, or Device Class Table index     |20
     +-------------------------------------------------+
 %25|  Virtual LDEV number of open spoolfile           |21
     +------------------------+------------------------+
 %26|  Volume table index     |   Sector address...    |22
     +------------------------+                        |
 %27|                             of spoolfile label.    |23
     +-------------------------------------------------+
 %30|   Number of extents     |////////////////////////|24
     +------------------------+------------------------+
 %31|              Last extent size (sectors)          |25
     +--+--+--+--+--+--+--+----------------------------+
 %32|SQ|//|RS|FD|SO|AB|//|      Number of copies       |26
     +--+--+--+--+--+--+--+----------------------------+
 %33|    Segment-relative link to next subentry,       |27
    |    this device or class. 0 ==> last subentry.    |
     +-------------------------------------------------+
 %34|   Number of records in spoolfile (doubleword)      |28
 %35|                                                  |29
     +------------------------+------------------------+
 %36|      Year MOD 100       | Julian Day of Year/2   |30
     +--+---------------+-----+----------+-------------+
 %37|DY| Hour (24 hr) |      Minute      | Seconds/4   |31
     +--+---------------+----------------+-------------+
Note:  Words 0-%24 are used in all subentries.  Words %25-%37,

       although present in all subentries, are zero unless the
       subentry is for a spooled file (spoolfile).

Word   0:  State -- State of subentry:
                      0 ==> Active
                      1 ==> Ready
                      2 ==> Open
                      3 ==> Locked
           CL    -- 1 ==> Word %24 is a class index  into  the
                          Device Class Table.
                    0 ==> Word %24 is the LDEV associated with
                          this subentry.
Word   1:  Type  -- Describes which  environment  created  the
                    subentry:
                      0 ==> Session' (SPOOK)
                      1 ==> Session
                      2 ==> Job
                      3 ==> Job'     (SPOOK)
Word %22:  IO    -- 1 ==> Output DFID
                    0 ==> Input  DFID
Word %23:  FS    -- There are one or more forms message re-
                    quests in the spoolfile.
           DA    -- The spoolfile was created via a :DATA record
                    (input spooling only).
           Head  -- The (segment-relative  address)/4  of  the
           index    head  entry  with  which  this subentry is
                    linked.  Since head entries are four words
                    long, this can be thought of as  an  index
                    into the head entry portion of the XDD--if
                    you disallow values of 0 and 1.
Word %24:        -- See description of Word 0.
Word %25   VDEV  -- LPDT index of virtual device  LDEV.  Simu-
                    lates the properties of a real LDEV to the
                    process which  FOPENs  a  new  (previously
                    non-existing)  file  (State field (XDD(0).
                    (1:2)) = 2 (Open)).
Word %26:  VTINX -- The volume table index of the logical  de-
                    vice  in  class SPOOL where the file label
                    (first extent) of the spoolfile lives.
Word %32:  SQ    -- 1 ==> Squeeze (purge) spoolfile  extents  as
                          the final copy is printed.  Obsolete
                          starting with C.00.20.
                    0 ==> Purge only when final copy printed.
           RS    -- 1 ==> Restart job when warmstarting (input
                          spooling only).
           FD    -- 1 ==> There are non-standard forms on  the
                          device.
           SO    -- Spaced Out bit.  File System could not ac-
                    quire a new extent when creating spoofile.
           AB    -- This is the $STDLIST of an aborted job.
Words %36-37:    -- Time stamp when spoofile was made READY, or
                    0D if not closed properly.  Julian day  is
                    9 bits starting with Word %36, bit 8.
```

SPOOK Tape Format

The overall format of output tapes produced by the SPOOK "OUTPUT" command is shown below. The various components of the tape are then described in detail. The format described here is subject to change as MPE evolves. Also, there may be errors in SPOOK which would cause the actual tape format to differ from the one described here in some cases. All numeric information is in integer format unless otherwise specified.

EOF

EOF

Label Record

EOF

File Directory Records

Device and Class Directory Record

EOF

Spoolfile

EOF

Spoolfile

EOF

.....

Mechanisms for End-of-tape and tape switching are the same as for STORE/RESTORE tapes.

Label Record

Words 0-13: "SPOOLFILETAPE LABEL-HP3000."

Word 23: reel number (first reel is number 1)

Word 24: date (from CALENDAR intrinsic)

Words 25&26: time (from CLOCK intrinsic)

Words 30&31: "MPE V" if an MPE V SPOOK tape

All other words are zero.

File Directory

The File Directory has one entry for each spoolfile on the tape. Each entry is 12 words, and entries are packed into as many 1020word records as needed. The last record will be padded with zeros if necessary. The entry format is:

Word 0: Device file id number (bit 0 is on to indicate that the file is an output spoolfile)

Words 1-3: zero

Words 4-7: User name

Words 8-11: Account Name

Device and Class Directory

The Device and Class Directory is contained in one 1024-word record. There is no EOF separating this record from the File Directory. This directory contains one entry for each logical device or device class linked to the spoolfiles on the tape. Also, there is an entry for each logical device in each class in the directory, whether or not that logical device was directly referenced by a spoolfile. The entries are packed into the tape record one after another in no particular order. The entry formats are shown below.

Logical Device Entry

Word 0: logical device number

Word 1: Bits 0:8 : device subtype
Bits 8:8 : 3 (=length of this entry in words)

Word 2: device type

Device Class Entry

Word 0: Device class number (negated). This is the number of the entry of this device class in the system's Device Class Table.

Word 1: Total number of words in this entry.

Words 2 on: The entire contents of the Device Class Table entry for this device class.

Spoolfile Format

```
ODD entry (32-word tape record)

Spoolfile block   --->  Two spoolfile blocks packed into one
Spoolfile block         1024-word tape record.

Two spoolfile blocks

Two spoolfile blocks

.....
```

The first few spoolfile blocks have been modified to contain user label information from the spoolfile. This is explained later.

Spoolfile Block Format

A spoolfile block is a 512-word block that contains variable length records in spooler format. Spoolfile records start at the first word of the block. The last record is followed by a -1 to indicate that no more records follow. The last two words of the block contain a doubleword which is the record number of the first record in the block.

Spoolfile Record Format

Word 0: Byte count of record - 2

Word 1: Byte count of data portion of record. Note that this count includes trailing blanks. However, trailing blanks are truncated in the actual record, so this count may be more than the number of bytes actually present in the data portion.

```
Word  2:     Function Code:  1=Fwrite
                             2=Fcontrol
                             3=Fopen
                             4=Fclose
               %100 and beyond=FDEVICECONTROL

Word  3:     P1 -- ATTACHIO parameter

Word  4:     P2 -- ATTACHIO parameter

Words 5 on:  Data Portion of Record
```

User Labels Information

Spoolfiles have a number of user labels with several kinds of information. These are:

1. Master: user label 0.
2. FOPEN entry catalog: user labels 1-10.
3. Circular queue for restart checkpointing: user labels 11-27.

Since older versions of MPE did not use user labels, a way was needed to incorporate them into the SPOOK tape format without losing forward and backward compatibility. The method used is to add several special spoolfile blocks to the beginning of the spoolfile on tape. Each of these blocks has exactly one FOPEN record at its beginning. This record is followed by a -1. Thus old versions of MPE will assume that the rest of the block is garbage. However, the rest of the block is actually used to contain user label information. The first two spoolfile blocks (i.e. the first tape record of the spoolfile proper) contain only the FOPEN records. The next 5 tape records actually contain user labels in addition to the FOPEN records. The user labels are packed 3 to a spoolfile block, 6 to a tape record. Each spoolfile block of 512 words has the following format:

```
Words       0-4:     FOPEN record

Word          5:     -1  (to "terminate" the block)

Words %200-%377:     user label

Words %400-%577:     user label

Words %600-%777:     user label
```

Following this special group of blocks, the spoolfile resumes a normal format. The special FOPEN records all have the number of user labels in P2.

It is often the case that some of the 27 user labels have not been initialized before the tape is written. In that case, their places will be filled with garbage. There is no easy way of detecting this except by careful inspection.

CHAPTER 15 UNIFIED COMMAND LANGUAGE (UNCL)

Reply Information Table (RIT)

DST %34; SIR %25

```
%|-------------------------------------------|
0|            NUMBER OF ENTRIES              | \
.flag=10
 |-------------------------------------------| |
1|          MAX NUMBER OF ENTRIES            | |
 |-------------------------------------------| |
2| POSITION OF NEXT FREE ENTRY SPACE IN QUEUE| | TABLE  57
 |-------------------------------------------| | HEADER wd
3|         NUMBER OF QUEUED ENTRIES          | |
 |-------------------------------------------| |
 | (52 WORDS TO HOLD PIN#'s OF QUEUED ENTRIES)| |
 |-------------------------------------------| |
 |                 UNUSED                    | |
 |===========================================| /
0|          PROCESS NUMBER (PIN)             |      \
 |-------------------------------------------|      |
1|            DST# (FOR REPLY)               |      |
 |-------------------------------------------|      |
2|       BUFFER ADDRESS (DST RELATIVE)       |      |
 |-------------------------------------------|      |
3| MAX LENGTH OF STRING | REPLY TYPE EXPECTED|      |
 |-------------------------------------------|      |
4|                                           |      |
 |-------------------------------------------|      |
5|                                           |      |
 |-------------------------------------------|      |
6|                                           |      | ENTRY
.flag=1
 |-------------------------------------------|      | (51
7|            # BYTES IN MESSAGE             |      |  wds)
 |-------------------------------------------|      |
 |                                           |      |
 |                                           |      |
 |             MESSAGE IN ASCII              |      |
 |                                           |      |
.flag=1
 |             (UP TO 86 CHARS.)             |      |
 |                                           |      |
 |                                           |      |
 |                                           |      /
 |-------------------------------------------|
```

```
NOTE:  Process Number = 0 means entry is empty
  Reply Type = 0 for number (num)
             = 1 for yes or no (y/n)
             = 2 for string (sxx)
             = 3 for yes, no, or STRING
```

```
.flag=2
             = 4 for string
  TABLE SIZE = 2046 words
.flag=2
  MAX # OF ACTIVE ENTRIES = 39
  MAX # OF QUEUED ENTRIES = 52
```

Message System General Description

The message system consists of the following parts:

- Callable intrinsic GENMESSAGE.
- Uncallable procedure GENMSG which is used by MPE.
- System message catalog (CATALOG.PUB.SYS) and any number of user catalogs.
- Program MAKECAT which builds message catalogs.
- MESSAGE SIR %24
- MESSAGE SYSGLOB CELLS %371-373
- MESSAGE DATA SEGMENT

The message system is used by calling GENMESSAGE (or GENMSG) with a message number. The message system fetches the message from a message catalog, inserts parameters, then routes the message to a file or returns the message in a buffer to the caller.

A message catalog is a numbered editor-type file containing sets of messages. The sets serve to break a catalog into manageable portions. A message system user may call GENMESSAGE using either his own message catalog or using MPE's catalog (CATALOG.PUB.SYS).

After creating a message file, run the program MAKECAT in order to build a catalog that is readable by the message system. This file is still readable by the editor (it can be "texted") but it contains a directory (written as a userlabel).

In order to use the message catalog, the program must first open the message catalog, then call GENMESSAGE with the file number, set number and message number. (MPE users don't need to open the catalog, GENMSG automatically uses CATALOG.PUB.SYS.) The file must be opened with the aoptions "NOBUF" and "MULTI" -record access.

Message Catalog

Messages in the catalog can be of any length and can contain up to five parameters. Continuation of a message is indicated by "%" or "&" at the end of a line. The "%" symbol indicates that the message is continued and that a carriage return, line feed be issued the terminal. The "&" symbol indicates that the message is continued on the same line with no carriage return, line feed.

Parameters may be inserted into the message fetched from the catalog. The parameters are passed in the GENMESSAGE (or GENMSG) call and inserted wherever a "!" is found. For the system message catalog, the back slash (\) is also a parameter, reflecting a logical device number. The message is routed to the user associated with that logical device through the :ASSOCIATE command. Message sets are indicated by "$SET n" starting in column 1 (the rest of the line is treated as a comment). Maximum value for n is 63. Comments can be inserted in the catalog by placing "$" in column 1. Message numbers are positive integers, need not be contiguous, but must be in ascending order. After processing by the program MAKECAT, the catalog file contains records of 80 bytes, blocked 16, in 32 extents. (The system message catalog is only one extent, however). The format of the message catalog is as follows:

```
   $SET 1    SYSTEM MESSAGES
   1 LDEV #! IN USE BY FILE SYSTEM
   2 LDEV #! IN USE BY DIAGNOSTICS
   3 LDEV IN USE, DOWN PENDING
   5 IS "!" ON LDEV#! (Y/N)?
      .
      .
      .
   $ MESSAGE 35 IS TWO LINES LONG, A PARAMETER STARTS THE
   $ FIRST LINE AND THE SECOND LINE IS "HP32002"
   35 !%
   HP32002B.00.!
      .
      .
      .
   276 LDEV # FOR "!" ON ! (NUM)!
   $
   $SET 2 CIERROR MESSAGES
   82 STREAM FACILITY NOT ENABLED:  SEE OPERATOR.  (CIERR 82)
   200 MORE THAN 30 PARAMETERS TO BUILD COMMAND.  (CIERR 200)
      .
      .
      .
   204 FILE COMMAND REQUIRES AT LEAST TWO PARAMETERS, INCLUDING
```

```
   THE
    FORMAL NAME OF THE FILE  (CIERR 204)
      .
      .
```

MAKECAT Program

The program MAKECAT.PUB.SYS is used to build message catalogs (and also HELP catalogs). The program's input file has the formaldesignator INPUT, which must be used for all entry points. The program has the following entry points:

(no entry point) - Reads from input file and builds a temporary file (formaldesignator CATALOG). Also renames any old temporary CATALOG, CATnn, using an archival numbering scheme (i.e., CAT1, CAT2, etc.).

BUILD - (Must log on under MANAGER.SYS.) Reads from input file, build the system message catalog (formaldesignator CATALOG), and installs the message system. Existing catalog is renamed CATnnnn according to the same scheme as for no entry point (above). Installation of the message system means moving the directory contained in the userlabel of the catalog into a data segment. The DST number and the disc address of CATALOG are placed in system global area. The message system may be installed while the system is running.

DIR - (Must have PM or OP capability.) Installs the system message catalog (does not build a new one). Opens input file, moves the directory in the CATALOG into a data segment, and places the DST number and disc address of CATALOG in system global area. This may be done when the message system seems to be "broken", but the catalog is intact. (MPE is issuing "MISSING MSG. SET=nn. MSG=nn" at terminals and at the console.) This may be done while the system is running.

HELP - Used to build the HELP catalog. Reads input file and builds a HELP catalog (formaldesignator HELPCAT).

Message System CATALOG.PUB.SYS

```
$SET 1  - System messages.
$SET 2  - CI errors and warnings messages.
$SET 3  - Miscellaneous ABORT messages.
$SET 4  - Program error abort messages.
$SET 5  - Intrinsics abort messages.
$SET 6  - Run-time abort messages.
$SET 7  - CI general messages.
$SET 8  - File System error messages.
$SET 9  - Loader error messages.
$SET 10 - CREATE error messages.
$SET 11 - ACTIVATE error messages.
$SET 12 - SUSPEND error messages.
$SET 13 - MYCOMMAND error messages.
$SET 14 - LOCKGLORIN error messages.
$SET 15 - Private Volumes error messages.
$SET 16 - DS/3000 messages.
$SET 17 - HELP facility error messages.
$SET 18 - Graphic devices messages.
$SET 19 - Serial Disc error messages.
$SET 20 - User Logging error messages.
$SET 21 - Association Utility (ASOCTABL) messages.
$SET 22 - 2680A Page Printer messages.
$SET 25 - 2680A Page Printer error file messages.
$SET 26 - Disc Free Space messages.
$SET 27 - System Internal Error messages.
```

Message Set Directory

```
   DST # IN SYSGLOB %373

   CAT DISC ADDR IN SYSGLOB %371-372

CREATED BY RUNNING MAKECAT.PUB.SYS.
KEPT IN A DATA SEGMENT AND IN A USER LABEL.

%             DATA SEGMENT              #
  |------------------------------------|
0 |        MAX. SET #                  |0 \        \
  |------------------------------------|  | HEADER  |
1 |     # OF MESSAGE RECORDS           |1 /         |
  |------------------------------------|            |
2 |  RECORD OFFSET TO FIRST MESSAGE    |2 \         |
  |------------------------------------|  | SET 1   | USER
3 |        FIRST MESSAGE #             |3 /         | LABEL
  |------------------------------------|            |
4 |  RECORD OFFSET TO FIRST MESSAGE    |4 \         |
  |------------------------------------|  | SET 2   |
5 |        FIRST MESSAGE #             |5 /         |
  |------------------------------------|            |
  |                                    |            |
           EMPTY ENTRY                              |
                                                    |
  |                                    |            |
  |------------------------------------|            |
50|  RECORD OFFSET TO FIRST MESSAGE    |40\         |
  |------------------------------------|  | SET 63  |
51|        FIRST MESSAGE #             |41/         |
  |------------------------------------|            |
52|              0                     |42\         |
  |------------------------------------|  | CUR MSG |
53| RECORD OFFSET TO CURRENT MESSAGE   |43/         |
  |------------------------------------|           /
54|           MESSAGE                  |44
  |            BUFFER
              (640 WORDS)

  |                                    |
  |------------------------------------|
1253                                 683

  EMPTY ENTRY:
  |------------------------------------|
  | RECORD OFFSET OF NEXT IN-USE SET   |
  |------------------------------------|
  |             -1                     |
  |------------------------------------|
```

HELP Subsystem

```
KEPT AS USER LABEL
READ ONTO USER'S STACK
USES SEARCH INTRINSIC FORMAT
VARIABLE ENTRY SIZE


%
 |----------------------------------------|
0|       DIRECTORY SIZE (WORDS)           |
 |----------------------------------------|
1| ENTRY LGTH (BYTES) | KEYWORD LGTH (BYTES) | \
 |----------------------------------------| |
2|               ENTRY                    | |
 |              KEYWORD                   | |  ENTRY
                                            |
                1-255 BYTES                 |
 |                                        | |
 |----------------------------------------| |
 |     ENTRY RECORD # IN CICAT            | |
 |   LEFT BYTE        |     RIGHT BYTE    | /
 |----------------------------------------|
 | ENTRY LGTH (BYTES) | KEYWORD LGTH (BYTES) | \
 |----------------------------------------| |
 |               ENTRY                    | |
                KEYWORD                     |  ENTRY
               1-255 BYTES                  |
 |                    |-------------------| |
 |                    | ENTRY REC # LEFT BYTE| |
 |----------------------------------------| |
 |ENTRY REC # R. BYTE | ENTRY LGTH (BYTES)  | / \
 |----------------------------------------|    |
 |KEYWORD LGTH (BYTES)|                    |    |
 |--------------------                     |    |
 |               ENTRY                     |    |
                KEYWORD                         |  ENTRY
               1-255 BYTES                      |
 |                                        |     |
 |----------------------------------------|     |
 |             ENTRY REC #                |     |
 |   LEFT BYTE        |     RIGHT BYTE    |     /
 |----------------------------------------|
 |                                        |


 |                                        |
 |----------------------------------------|
```

UDC Directory

```
*EXTRA DATA SEGMENT - DST # IN DB+%255 OF UMAIN STACK

*BUILT BY INITUDC

 0  1  2  3     6 7 8           15
|-----------------------------------|
|LT|LN|NH|NB|   |TY |   ENTRY SIZE  | \ LT-OPTION LIST
|-----------------------------------| | LN-OPTION LOGON
|      HEADER RECORD NUMBER         | | NH-OPTION NOHELP
|-----------------------------------| | NB-OPTION NOBREAK
|        BODY RECORD NUMBER         | | TY- 00=USER UDC
|-----------------------------------| |     01=ACCOUNT UDC
|    FILE NUMBER    |  COMMAND LENGTH | |     10=SYSTEM UDC
|-----------------------------------| |
|                                   | > ENTRY
|              COMMAND              | |
                NAME                  |
             (1-16 BYTES)             |
|                                   | |
|                                   | /
|-----------------------------------|
|                                   |
|                                   |
              ENTRIES

|                                   |
|                                   |
|-----------------------------------|
|                   |       0       |  ENTRY SIZE=0 ENDS
|-----------------------------------|        DIRECTORY
```

UDC's COMMAND.PUB.SYS

*RECORD SIZE = 20(10) WORDS, 6 RECORDS/BLOCK

*KEEPS TRACK OF WHO IS USING WHAT UDC CATALOG

*CAN BE PURGED TO DISABLE UDC'S

*CAN BE REBUILT TO RE-ENABLE UDC'S

```
%      RECORD 0      #       %    FREE ENTRY      #
 |-----------------|          |-----------------|
0|1st FREE ENTRY # |0        0|NEXT FREE ENTRY #|0
 |-----------------|          |-----------------|
1|    not used     |1        1|  ENTRY TYPE=0   |1
 |-----------------|          |-----------------|
2|   MAX IN USE    |2        2|                 |2
 |-----------------|          |                 |
3|    # IN USE     |3              not used
 |-----------------|
4|                 |4         |                 |
 |    not used     |          |                 |
                              |                 |
 |                 |          |                 |
23|                |19      23|                 |19
 |-----------------|          |-----------------|
```

COMMAND.PUB.SYS (Cont.)

```
 %  USER ENTRY       #       %  FREE ENTRY        #
  |-----------------|         |-----------------|
 0| CATALOG ENTRY # |0       0|NEXT CAT. ENTRY #|0
  |-----------------|         |-----------------|
 1|  ENTRY TYPE=1   |1       1| ENTRY TYPE = 2  |1
  |-----------------|         |-----------------|
 2|                 |2       2|                 |2
  |      USER*      |         |   FILE NAME     |
 3|                 |3       3|                 |3
  |                 |         |  FOPEN FORMAT:  |
 4|                 |4       4|                 |4
  |                 |         |                 |
 5|                 |5       5|                 |5
  |-----------------|         |                 |
 6|                 |6       6|      FILE       |6
  |                 |         |                 |
 7|    ACCOUNT*     |7       7|  [/LOCKWORD]    |7
  |                 |         |                 |
10|                 |8      10|     GROUP       |8
  |                 |         |                 |
11|                 |9      11|    ACCOUNT      |9
  |-----------------|         |                 |
12|                 |10     12|       0         |10
  |                 |         |                 |
13|    not used     |11     13|                 |11
  |                 |         |                 |
14|                 |12     14|                 |12
  |                 |         | (UP TO 36 BYTES)|
15|                 |13     15|                 |13
  |                 |         |                 |
16|                 |14     16|                 |14
  |                 |         |                 |
17|                 |15     17|                 |15
  |                 |         |                 |
20|                 |16     20|                 |16
  |                 |         |                 |
21|                 |17     21|                 |17
  |                 |         |                 |
22|                 |18     22|                 |18
  |                 |         |                 |
23|                 |19     23|                 |19
  |-----------------|         |-----------------|
```

* IF THE USER FIELD AND THE ACCOUNT FIELD CONTAIN "@_______", THIS INDICATES SYSTEM LEVEL UDC'S.

IF ONLY THE USER FIELD CONTAINS @ AND 7 SPACES, THIS INDICATES ACCOUNT LEVEL UDC'S.

CI Stack Definition

```
DB+%0      |  BCOMIMAGE (Byte Ptr. To Command)   |
           |-------------------------------------|
DB+%1      |           COMMAND IMAGE             |
          \|            (280 bytes)             \|
           \                                     \
           |\                                    |\
           |-------------------------------------|
DB+%215    |            LINELENSTACK             |
          \|             (30 words)             \|
           \                                     \
           |\                                    |\
           |-------------------------------------|
DB+%253    |     NEXTMSG (Not currently used)    |
           |-------------------------------------|
DB+%254    |            THIS IS SPARE            |
           |-------------------------------------|
DB+%255    |                UDC0                 |
           |-------------------------------------|
DB+%256    |                UDC1                 |
           |-------------------------------------|
DB+%257    |                UDC2                 |
           |-------------------------------------|
DB+%260    |                UDC3                 |
           |-------------------------------------|
DB+%261    |                UDC4                 |
           |-------------------------------------|
DB+%262    |              IFNESTING              |
           |-------------------------------------|
DB+%263    |               IFSKIP                |
           |-------------------------------------|
DB+%264    |              ELSESEEN               |
           |-------------------------------------|
DB+%265    |               CIFLAGS               |
           |-------------------------------------|
DB+%266    |         CONTINUE STATE STACK        |
           |             (2 words)               |
           |-------------------------------------|
DB+%270    |            PENDINGCOMLEN            |
           |-------------------------------------|
DB+%271    |      BLASTCOMIMAGE (Byte Ptr.)      |
           |-------------------------------------|
DB+%272    |          LAST COMMAND IMAGE         |
          \|            (280 bytes)             \|
           \                                     \
           |\                                    |\
           |-------------------------------------|
```

Field Definitions

BCOMIMAGE: Byte pointer to COMIMAGE (sometimes called WCOMIMAGE) in the CI stack.

COMMAND IMAGE: Command character string currently being executed.

LINELENSTACK: A CI command can span up to 30 input lines. This stack holds the length of each input line.

NEXTMSG: Used to be used to link messages together. No longer being used.

THIS IS SPARE: Not used.

UDC0: Holds the DST number of the UDC definitions.

UDC1: Holds the old S register value for UDC's.

UDC2: (0:1)--FLUSHUDC, used by :SETCATALOG

UDC3: UDC options for current UDC.

UDC4: (0:1)--UDC Fatal Ci Error
       (1:1)--UDC EXITBREAK
       (2:1)--UDC BREAKDETECTED
       (3:1)--UDC NOPRINT
       (4:1)--UDC IMAGEADJUST
       (10:6)--UDC NESTLEVEL

IFNESTING: Level of nesting of :IF commands.

IFSKIP: Whether the current commands are being skipped as the false part of a :IF command.

ELSESEEN: Level of the :ELSE commands.

CIFLAGS: (13:1)--Sequenced: line numbers at rear.
          (15:1)--Not REDOable (last command).

CONTINUE STATE STACK: History of the :CONTINUE commands.
      = 0--no :CONTINUE
      = 1--just seen
      = 2--in effect.

PENDINGCOMLEN: If <> 0, command is already in stack and this word is the command string length.

BLASTCOMIMAGE: Byte pointer to last command image.

LAST COMMAND IMAGE: When a command completes execution, the command string is copied here for use by the :REDO command.

Association DST Layout

```
|========================================|
|                                        |  0     DST %42
|                                        |  1
|               Not                      |  2     SIR %30
|                                        |  3
|               Used                     |  4
|                                        |  5     One entry/
|                                        |  6     system ldev
|========================================|
|  JMAT Index                            |  7   \
|----------------------------------------|      |
|  JIT DST Number                        |  8   |
|----------------------------------------|      |
| DST rel. index to user's next entry.   |  9   |- Ldev 1
|----------------------------------------|      |
                                             (Associated)
|  Class name under which this ldev is   | 10   |
|  associated.  Left justified and       | 11   |
|  padded with blanks.  8 bytes.         | 12   |
|                                        | 13   /
|========================================|
|                  0                     | 14   \
|----------------------------------------|      |
|                  0                     | 15   |
|----------------------------------------|      |
|                  0                     | 16   |- Ldev 2
|----------------------------------------|      |
                                             (Unassociated
                                             )
|                                        | 17   |
|               Don't                    | 18   |
|               Care                     | 19   |
|                                        | 20   /
|========================================|
|                                        |
.                  .                     .
.                  .                     .
.                  .                     .
|                                        |
|========================================|
|         JMAT Index or 0                | 7*n  \
|----------------------------------------|      |
|          JIT DST Number or 0           |      |
|----------------------------------------|      |
|       Next Entry Pointer or 0          |      |- Ldev n
|----------------------------------------|      |
| Classname under which LDEV is          |      |
| associated or undefined.               |      |
|                                        |      /
|========================================|
```

CHAPTER 16 SYSDUMP/INITIAL

CONFDATA File

Record 0 of CONFDATA File (CTABO)

```
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0|               CHECKSUM OF CTAB                |0
   |-----------------------------------------------|
  1|           CURRENT VERSION OF CTAB             |1
   |-----------------------------------------------|
  2|             STANDARD STACK SIZE               |2
   |-----------------------------------------------|
  3|            CORESIZE IN K WORDS                |3
   |-----------------------------------------------|
  4|           TERMINAL BOUND PRIORITY             |4
   |-----------------------------------------------|
  5|               NORMAL PRIORITY                 |5
   |-----------------------------------------------|
  6|             CPU BOUND PRIORITY                |6
   |-----------------------------------------------|
  7|           # OF SECONDS TO LOG-ON              |7
   |-----------------------------------------------|
 10|        LOG FILE RECORD SIZE (SECTORS)         |8
   |-----------------------------------------------|
 11|           LOG FILE SIZE (RECORDS)             |9
   |-----------------------------------------------|
 12|///////////////////////////////////////////////|10
   |-----------------------------------------------|
 13|           LOG BITS (ONLY 11 USED)             |11
   |                                               |
 14|                                               |12
 15|      <<DEFINES WHAT IS BEING LOGGED>>         |13
 16|                                               |14
   |                                               |
 17|                                               |15
   |-----------------------------------------------|
 20|      DEFAULT JOB/SESSION CPU TIME LIMIT       |16
   |-----------------------------------------------|
   . ///////////////////////////////////////////// .
   . ///////////////////////////////////////////// .
   . ///////////////////////////////////////////// .
   |-----------------------------------------------|
 34|          MAXIMUM OPEN SPOOL FILES             |28
   |-----------------------------------------------|
 35|///////////////////////////////////////////////|29
   |-----------------------------------------------|
 36|                                               |30
   |   MAXIMUM # OF SPOOL FILES (KILO SECTORS)     |
 37|                                               |31
   |-----------------------------------------------|
 40|///////////////////////////////////////////////|32
   |-----------------------------------------------|
 41|         # SECTORS PER SPOOL EXTENT            |33
   |-----------------------------------------------|
```

Record 1 of CONFDATA File (CTAB)

```
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0|               # OF CST ENTRIES                |0
   |-----------------------------------------------|
  1|               # OF DST ENTRIES                |1
   |-----------------------------------------------|
  2|               # OF PCB ENTRIES                |2
   |-----------------------------------------------|
  3|               # OF IOQ ENTRIES                |3
   |-----------------------------------------------|
  4|             # OF TERMINAL BUFFERS             |4
   |-----------------------------------------------|
  5|          # OF CST EXTENSION ENTRIES           |5
   |-----------------------------------------------|
  6|  INTERRUPT CONTROL STACK SIZE (Q1 to Z1)      |6
   |-----------------------------------------------|
  7|         # UCOP REQUEST QUEUE ENTRIES          |7
   |-----------------------------------------------|
 10|             # BREAKPOINT ENTRIES              |8
   |-----------------------------------------------|
 11|                # TRL ENTRIES                  |9
   |-----------------------------------------------|
 12|                # LOCAL RINS                   |10
   |-----------------------------------------------|
 13|                # GLOBAL RINS                  |11
   |-----------------------------------------------|
 14|              # OF SYSTEM BUFFERS              |12
   |-----------------------------------------------|
 15|            # OF CONCURRENT PROGS              |13
   |-----------------------------------------------|
 16|          LOADER SEGMENT SIZE                  |14
   |-----------------------------------------------|
   .///////////////////////////////////////////// .
   .///////////////////////////////////////////// .
   .///////////////////////////////////////////// .
   |-----------------------------------------------|
 24|             SIZE OF VIRTUAL MEMORY            |20
   |-----------------------------------------------|
 25|          DIRECTORY SIZE (SECTORS)             |21
   |-----------------------------------------------|
   .///////////////////////////////////////////// .
   .///////////////////////////////////////////// .
   .///////////////////////////////////////////// .
```

CONDATA (Cont.)

```
   |-----------------------------------------------|
 36|          MAXIMUM CODE SEGMENT SIZE            |30
   |-----------------------------------------------|
 37|     MAXIMUM # OF CODE SEGMENTS/PROCESS        |31
   |-----------------------------------------------|
 40|        MAXIMUM STACK SIZE (MAXDATA)           |32
   |-----------------------------------------------|
 41|      MAXIMUM EXTRA DATA SEGMENT SIZE          |33
   |-----------------------------------------------|
 42|  MAXIMUM # OF EXTRA DATA SEGMENTS/PROCESS     |34
   .///////////////////////////////////////////// .
   .///////////////////////////////////////////// .
   .///////////////////////////////////////////// .
   |-----------------------------------------------|
 50|         MAXIMUM # RUNNING SESSIONS            |40
   |-----------------------------------------------|
 51|         MAXIMUM # OF RUNNING JOBS             |41
   |-----------------------------------------------|
 52|              # LOG PROCS                      |42
   |-----------------------------------------------|
 53|                LOG ID's                       |43
   |-----------------------------------------------|
 54|        # DISC REQUEST TABLE ENTRIES           |44
   |-----------------------------------------------|
 55|       # SPECIAL REQUEST TABLE ENTRIES         |45
   |-----------------------------------------------|
 56|       # PRIMARY MESSAGE TABLE ENTRIES         |46
   |-----------------------------------------------|
 57|         # SWAP TABLE ENTRIES                  |47
   |-----------------------------------------------|
 58|        # SECONDARY MESSAGE TABLE ENTRIES      |48
   |-----------------------------------------------|
```

DEVDATA.PUB.SYS

Overview

```
|-------------------------|
|       PARAMETERS        |
|-------------------------|
|      DRIVER TABLE       |
|-------------------------|
|          LPDT           |
|-------------------------|
|          LDT            |
|-------------------------|
|          LDTX           |
|-------------------------|
|   CLASS/TERM HEADER     |
|-------------------------|
|          CLASS          |
|-------------------------|
|        TERM DEF         |
|-------------------------|
|     ADD'L DVR TABLE     |
|-------------------------|
|         CS DEF          |
|-------------------------|
|        CS TABLE         |
|-------------------------|
```

Parameter Record

```
    |-------------------------|
 0  |        CHECKSUM         |
    |-------------------------|
 1  |         VERSION         |
    |-------------------------|
 2  |       NEXT RECORD       |
    |-------------------------|
 3  |      HIGHEST LDEV       |
    |-------------------------|
 4  |       HIGHEST DRT       |
    |-------------------------|
 5  |    NR. ADD'L DRIVERS    |
    |-------------------------|
    |--------------------------|
64  |          REC #           |  DVR TABLE
    |--                      --|
    |         LENGTH           |
    |--------------------------|
66  |          REC #           |  LPDT
    |--                      --|
    |         LENGTH           |
    |--------------------------|
68  |          REC #           |  LDT
    |--                      --|
    |         LENGTH           |
    |--------------------------|
70  |          REC #           |  LDTX
    |--                      --|
    |         LENGTH           |
    |--------------------------|
72  |          REC #           |  DCTH
    |--                      --|
    |         LENGTH           |
    |--------------------------|
74  |          REC #           |  CLASS
    |--                      --|
    |         LENGTH           |
    |--------------------------|
76  |          REC #           |  TERM DEF
    |--                      --|
    |         LENGTH           |
    |--------------------------|
78  |          REC #           |  ADD'L DVR
    |--                      --|
    |         LENGTH           |
    |--------------------------|
80  |          REC #           |  CS DEF
    |--                      --|
    |         LENGTH           |
    |--------------------------|
82  |          REC #           |  CS TABLE
    |--                      --|
    |         LENGTH           |
    |--------------------------|
```

Driver Table

The Driver Table consists of 7 word entries, in correspondence to the LDEV entries, up to the highest LDEV used, entry zero is a dummy entry.

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
 |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 |                     DRT #                     |
 |--|--------|--------|--|-----------------------|
 |CR| CHAN # |        |DS|       UNIT #          |
 |--|--------|--------|--|-----------------------|  TYPICAL ENTRY
 |                  MASTER LDEV                  |  FORMAT
 |-----------------------------------------------|
 |          D           |          R             |
 |----------------------|------------------------|
 |          I           |          V             |
 |----------------------|------------------------|
 |          N           |          A             |
 |----------------------|------------------------|
 |          M           |          E             |
 |----------------------|------------------------|
```

DS	DS DEVICE (if set DRT is zero)
CR	CORE RESIDENT
CHAN #	CHANNEL #
MASTER LDEV	LDEV of device which this DS device is linked to.

Words 3-7 contain the driver name.

SYSDUMP Format

```
  |------------------------------------| <---ENTRY POINT #1 (ROM BASED
  |             CHECKSUM               | 0   MACHINES)
  |        AMIGO CHANNEL PROGRAM       |
--|          WCS TABLE PRT             | 95
| |------------------------------------| 127
| |--            AMIGO               --|
| |------------------------------------|
->|            WCS TABLE               |
  |------------------------------------|
  |            WCS #1                  |
  |------------------------------------|
  |            WCS #2                  | Only for the 64/68. Refer to the
  |------------------------------------| WCS Table for the 64/68 below.
  |            WCS #n                  |
  |------------------------------------| <---ENTRY POINT #2 (WCS BASED
  |            CHECKSUM                | 0   MACHINES)
  |            AMIGO                   |
  |------------------------------------| 127
  |--                                --|
  |--          AMIGO                 --|
  |------------------------------------|
  |              ICS                   |
  |------------------------------------|
  |            LOW CORE                |
  |------------------------------------|
  |     Initial       CST              |
  |------------------------------------|
  |            CS TABLE                |
  |------------------------------------|
  |      DEVICE CLASS TABLE HEADER     |
  |------------------------------------|
  |        DEVICE CLASS TABLE          |
  |------------------------------------|
  |     TERMINAL DESCRIPTOR TABLE      |
  |------------------------------------|
  |              VTAB                  |
  |------------------------------------|
  |             OLDVTAB                | *
  |------------------------------------|
  |  DISC COLD LOAD INFORMATION TABLE  | *
  |------------------------------------|
  |              CTAB                  |
  |------------------------------------|
  |              CTABO                 |
  |------------------------------------|
  |  COMMUNICATION RECORD              |
  |------------------------------------|
  |              CSDVR                 |
  |------------------------------------|
  |              CSDEF                 |
  |------------------------------------|
  |         INITIAL'S DB AREA          |
  |------------------------------------|
```

```
|            STACK MARKER              |
|--------------------------------------|
|         DRIVER TABLE                 |
|--------------------------------------|
|              LPDT                    |
|--------------------------------------|
|              LDT                     |
|--------------------------------------|
|              LDTX                    |
|--------------------------------------|
|         INITIAL'S SEGMENTS           |
|--------------------------------------|
|              RIN TABLE               | *
|--------------------------------------|
|     LOGGING IDENTIFIER TABLE         | *
|--------------------------------------|
|         DIRECTORY HEADER             | *
|--------------------------------------|
|            DIRECTORY                 | *
|--------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|--------------------------------------|
| SYSTEM PROGRAMS, SL, NON-STD. DRIVERS|
|--------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|--------------------------------------|
|       STORE/RESTORE HEADER           |
|--------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|--------------------------------------|
|      STORE/RESTORE DIRECTORY         | *
|--------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|--------------------------------------|
|  USER FILES (SEPARATED BY "EOF's"    | *
|--------------------------------------|
|       STORE/RESTORE TRAILER          |
|--------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|--------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|--------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXX|
|--------------------------------------|
```

* NOT DUMPED IF DATE = CARRIAGE RETURN

NOTE: ON DISC, READ-SIO-PROGRAM KEPT IN DISC LABEL.

WSC Table Format

```
|-----------------------------------------|  0
| # Records to WCS                        |
| # Records of WCS                        |
| # Records after WCS                     |
|  WCS Record Size on Tape                |
|-----------------------------------------|  1
|                                         |
|-----------------------------------------|  2
|                                         |
|-----------------------------------------|  3
|                                         |
|-----------------------------------------|  4
|                                         |
|-----------------------------------------|
~                                         ~
|-----------------------------------------|
|                                         |
|-----------------------------------------|
```

Note: Currently only one entry used (Entry 4, by Series 64).

Series 64/68 WCS TABLE FORMAT

```
|---------|----------------|---------------|
|128 Word | WCS            |    LUT        |
|Header   |                |               |
|---------|----------------|---------------|


|-----------------------------------------|  0
|    Microcode Version (8 Bytes ASCII)    |
|-----------------------------------------|  4
| # of WCS LOCATIONS (64 Bit Words)       |
|-----------------------------------------|  6
| # of LUT LOCATIONS (32 Bit Words)       |
|-----------------------------------------|  8
|        WCS CHECKSUM                     |
|-----------------------------------------|  8
|        LUT CHECKSUM                     |  9
|-----------------------------------------|
```

Store Tape Format

First Volume

```
|------------------------------------------|
|XXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXXX|
|------------------------------------------|
|XXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXXX|
|------------------------------------------|
|        "STORE/RESTORE LABEL -            |0     \
|              HP/3000."                   |13    |
|------------------------------------------|      |
|                "VIIB"                    |14    |
|                                          |15    |
|------------------------------------------|      |
|       PARTIAL FIRST FILE FLAG            |16    |
|------------------------------------------|      |
|               CHECKSUM                   |17    |
|------------------------------------------|      |
|       DIRECTORY INDEX OF FIRST FILE      |18    |
|------------------------------------------|      | HEADER
|                                          |19    |  40 WORDS
|                                          |      |
|                                          |      |
|                                          |22    |
|------------------------------------------|      |
|             VOLUME NUMBER                |23    |
|------------------------------------------|      |
|                 DATE                     |24    |  DATE:
|------------------------------------------|      |   0:7 last 2 digits
|                 TIME                     |25    |       of year
|                                          |26    |   7:9 Julian date
|------------------------------------------|      |
|  TAPEBLOCKSIZE (#WORDS/BLOCK;def=4096)   |27    |  TIME:
|------------------------------------------|      |   25.(0:8) hours
|                                          |28    |      (8:8) minutes
|                                          |      |   26.(0:8) seconds
|                                          |      |      (8:8) .1 secs.
|                                          |      |
|                                          |39    /
|------------------------------------------|
```

First Volume (Cont.)

```
|------------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXX|
|------------------------------------------|
|                    .                     |            \
|                    .                     |            |
|                    .                     |            |
|------------------------------------------|            |
|                FILE NAME                 |\           |
|------------------------------------------||TYP FILE   |
|               GROUP NAME                 || ENTRY     | VOLUME
|------------------------------------------||(12 WDS.)  | DIRECTORY:
|               ACCT. NAME                 |/           |  # ENTRIES
|------------------------------------------|            | DETERMINED
|                    .                     |            | BY TAPEBLOCK-
|                    .                     |            | SIZE
|                    .                     |            /
|------------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXX|
|------------------------------------------|
|                                          |            \
|      FILES  (separated by "EOF's")       |            | FILES
|                                          |            |
                                                        /
```

Subsequent Volumes

```
|------------------------------------------|
|        "STORE/RESTORE LABEL-             |0             \
|              HP/3000."                   |13            |
|------------------------------------------|              |
|                "VIIB"                    |14            |
|                                          |15            |
|------------------------------------------|              |
|       PARTIAL FIRST FILE FLAG            |16 FLAG=1:    |
|------------------------------------------|   1st FILE   |
|               CHECKSUM                   |17 ON THIS    |
|------------------------------------------|   VOL IS A   |
|      DIRECTORY INDEX OF FIRST FILE       |18 PARTIAL.   |  HEADER
|------------------------------------------|              |   40 WDS.
|                                          |19            |
|                                          |22            |
|------------------------------------------|              |
|             VOLUME NUMBER                |23            |
|------------------------------------------|              |
|                 DATE                     |24            |
|------------------------------------------|              |
|                 TIME                     |25            |
|                                          |26            |
|------------------------------------------|              |
|            TAPEBLOCKSIZE                 |27            |
|------------------------------------------|              |
|                                          |28            |
|                                          |39            / NOTE: NO EOF.
|==========================================|
|                   .                      |              \
|                   .                      |              |
|                   .                      |              |
|------------------------------------------|              |
|               FILE NAME                  |\             |
|------------------------------------------||TYPICAL      |
|              GROUP NAME                  || FILE         | VOLUME
|------------------------------------------|| ENTRY        | DIRECTORY
|               ACCT NAME                  |/              |
|------------------------------------------|              |
|                   .                      |              |
|                   .                      |              |
|                   .                      |              /
|------------------------------------------|
|XXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXXX|
|------------------------------------------|
|                                          |              \
|                <FILES>                   |              | FILES
|         (separated by "EOF's)            |              |
                                                          /
```

End of Volume

```
                                                         \
|                                          |             |
|                 <FILES>                  |             |
|          (separated by "EOF's)           |             | FILES
|                                          |             |
|                                          |             /
|------------------------------------------|
XXXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXX|
|------------------------------------------|
|     "STORE/RESTORE LABEL-HP/3000."       |0            \
|                                          |13           |
|------------------------------------------|             |
|                                          |14           |
|                                          |             |
|                                          |20           |
|------------------------------------------|             |
|   FLAG: PRECEDING EOF MARKS FILE ENDED   |21           | TRAILER
|------------------------------------------|             |
|  FLAG: PRECEDING EOF MARKS TAPESET ENDED |22           |  40 WDS.
|------------------------------------------|             |
|               VOLUME NO.                 |23           |
|------------------------------------------|             |
|                   DATE                   |24           |
|------------------------------------------|             |
|                   TIME                   |25           |
|                                          |26           |
|------------------------------------------|             |
|                                          |27           |
|                                          |             |
|                                          |39           /
|------------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXX|
|------------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXX|
|------------------------------------------|
|XXXXXXXXXXXXXXXXXXX EOF XXXXXXXXXXXXXXXXXX|
|------------------------------------------|
```

CHAPTER 17 MISCELLANEOUS

Labeled Tape Subsystem

The MPE labeled tape subsystem permits convenient access to tapes labeled to either ANSI or IBM standards. It operates as a set of subprocedures to the file system. A labeled tape consists of one or more logical files. Each logical file consists of three physical files, i. e. tape areas delimited by tapemarks. The first physical file contains header labels, the second contains the data, and the third contains trailer labels which are (except for minor differences) copies of the header labels. The tape mark following trailer labels will be followed either by header labels for the next file, or by another tapemark if there is no next file. Labels are 80 bytes long, and conventionally are identified by their first four characters (three letters and a digit) and contain information as follows (CP := character position; L:= length):

VOL1: Present only on the first file of a volume, the volume label contains the volume identifier, which is usually the number on the tape strap, and is thus not expected to be changed.

CP	Field Name	L	Content
1/3	Label identifier	3	"VOL"
4	Label Number	1	"1"
5/10	Volume Identifier	6	Vol ID
11	Accessibility	1	"0" if IBM, else " "
12/79	Not used	62	Blanks
80	Label-Standard Version	1	"1" if HP ANSI else " "

UVLn: User volume labels. May be present on tapes from foreign shops, but are not written by MPE. If encountered, they are ignored.

HDR1: First header label. Required for each file. Specifies:

CP	Field Name	L	Content
1/3	Label identifier	3	"HDR"
4	Label Number	1	"1"
5/21	File Identifier	17	File name, if tape was not written by MPE, only the first eight are significant.
22/27	Volume Set Identifier	6	Names the volume on which the set of files begins
28/31	Reel Number	4	Counts the reels that contain this file (1 starts)
32/35	File sequence number	4	Counts the files in the set of files (1 starts)
36/41	Not Used	6	MPE writes blanks
42/47	Creation Date	6	Year and day within year when the file was written.
48/53	Expiration Date	6	Year and day within year when the file may be over-written without permission.
54	Accessibility	1	%230 if Lockword, "0" if IBM
55/60	Block count	6	Number of blocks if IBM.
61/73	System Code	13	"HP MPE 3000 "
74/80	Not Used	7	Blanks

HDR2: Second header label. Although defined by the standard, may be missing on foreign tapes. Contains:

CP	Field Name	L	Content
1/3	Label identifier	3	"HDR"
4	Label Number	1	"2"
5	Record Format	1	"F" = Fixed "V" = Variable "U" = Undefined Others treated as Undefined
6/10	Block Length	5	Block length (in character format).
11/15	Record Length	5	Record length (adhering to to MPE rules) in characters.
16/23	Lockword	8	MPE File Lockword.
24/36	Not Used	13	MPE writes blanks
37	Record Type	1	"A" = ASCII "B" = Binary.
38	Carriage Control	1	"C" = control " " = no control.
39/80	Not Used	42	Blanks

IBM has a slightly different format. It is:

CP	Field Name	L	Content
1/3	Label identifier	3	"HDR"
4	Label Number	1	"2"
5	Record Format	1	"F" = Fixed "V" = Variable "U" = Undefined Others treated as Undefined
6/10	Block Length	5	Block length (in character format).
11/15	Record Length	5	Record length (adhering to to MPE rules) in characters.
16	Not Used	1	Blank.
17	IBM Position	1	"0" = no volume switch "1" = a switch has occurred.
18/38	Not Used	11	Blanks.
39	IBM Block Attribute.	1	"B" = Blocked records. "S" = Spanned records. "R" = Blocked and Spanned. " " = No blocked or spanned.
40/80	Not Used	41	Blanks

User header labels: optional. Standard prescribes UHLn in the first four characters, but MPE doesn't care.

EOV1: End of Volume; used as first trailer label. Required if the logical file is continued onto another reel. Identical to HDR1, except contains the number of physical blocks of data in the data area.

CP	Field Name	L	Content
1/3	Label identifier	3	"EOV"
4	Label Number	1	"1"
5/54	Same as HDR1	50	
55/60	Block Count	6	Number of data blocks since last beginning of file section label group.
61/80	Same as HDR1	20	

EOV2: Defined by the standard, but may be missing on foreign tapes. Follows EOV1; format same as HDR2.

EOF1: End of File; used as first trailer label. Required if this is the end of the logical file. Format same as EOV1.

EOF2: Same as EOV2 except used after EOF1.

User trailer labels: optional. Standard prescribes UTLn in the first four characters, but MPE again doesn't care.

Tape Label Table

The tape label table is the private playground of the tape label subsystem. It consists of two parts: LDEV Control Blocks (LCBs) and Volume Control Blocks (VCBs). The LDEV area is set up at system initialization and contains one entry for each magnetic tape LDEV and serial disc device in the system. As is common in MPE, the first entry is a dummy which tells where the other things in the table are. The volume area contains one entry for each labeled tape volume requested or active on the system.

Although table entries are stored in an extra data segment, they are generally manipulated via local copies on the stack. The procedures GETLDEV and GETFNUM look for LDEV and volume entries as specified; they copy them to stack buffers and return the DST address for use in copying them back. POSTVTENT copies the entries back, and in the case of a new volume entry, allocates space for it in the volume section of the tape label table.

Initial will build the "uninitialized" TLT as follows:

```
 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
-----------------------------------------------------------------
|     Size of the table, in words (always > 1)                  |  0
|---------------------------------------------------------------|
|     Number of LDEVS in the table = X                          |  1
|---------------------------------------------------------------|
.flag=1
|                          LDEV#                           |T   |  2
|---------------------------------------------------------------|
|                                                               |

            Total of LDEVS (X) entries of above

|                                                               |
|---------------------------------------------------------------|
|                          LDEV#                           |T   |  X+2
|---------------------------------------------------------------|
|
          Expansion area
            during SETUP'TAPES
|                                                               |
-----------------------------------------------------------------
```

T: 1 if Tape drive 0 if not Tape drive (i.e. serial disc)

During PROGEN, SETUP'TAPES is called to initialize the table. The overall structure of the initialized TLT is:

```
TLTDST -- %32,#26                    TLTSIR -- %47,#39

 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
-----------------------------------------------------------------
|     Table initialization word (=1 when initialized)           |  0
|---------------------------------------------------------------|
|     Entry size (ESIZE) = %32,#26                              |  1
|---------------------------------------------------------------|
|   Table relative pointer to base of LCB entries (LTBASE) (1)  |  2
|---------------------------------------------------------------|
|   Table relative pointer to base of VCB entries (VTBASE) (2)  |  3
|---------------------------------------------------------------|
|   Table relative pointer to top of Volume table (VTTOP) (3)   |  4
|---------------------------------------------------------------|
|     Size of Tape Label Table, in words (VTMAX)                |  5
|---------------------------------------------------------------|
|                                                               |  6
|                                                               |  7
|                                                               |  10
|
                        not used
|                                                               |
|                                                               |  30
|                                                               |  31
|---------------------------------------------------------------|
|                                                               |  32
|                                                               |  <-(1)
      LDEV Control Block area -- one entry/mag tape drive
|                                                               |
|---------------------------------------------------------------|
|                                                               |  <-(2)
      Volume Control Block table -- contains VCB entries
                                    and free entries
|                                                               |
|---------------------------------------------------------------|
|                                                               |  <-(3)
      Area available for expansion of VCB table
|                                                               |
|---------------------------------------------------------------|
```

The LCB entries have the following structure:

```
 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
-----------------------------------------------------------------
|               | Type   | T | L | B | HP|                      |  0
|---------------------------------------------------------------|
|                   Logical device number                       |  1
|---------------------------------------------------------------|
|                        VCB address                            |  2
|---------------------------------------------------------------|
|                        Reel number                            |  3
|---------------------------------------------------------------|
|                   File sequence number                        |  4
|---------------------------------------------------------------|
|                       Creation date                           |  5
|---------------------------------------------------------------|
|                      Expiration date                          |  6
|---------------------------------------------------------------|
|                                                               |  7
|                         File name                             |  10
|

|                                                               |
|                                                               |  16
|                               +-------------------------------|
|                               |                               |  17
+---------------------------------------------------------------+
|                                                               |  20
|                                                               |  21
|                        (not used)                             |  22
|                                                               |  23
|---------------------------------------------------------------|
|                                                               |  24
|                   Volume set identifier                       |  25
|                                                               |  26
|---------------------------------------------------------------|
|                                                               |  27
|                     Volume identifier                         |  30
|                                                               |  31
|---------------------------------------------------------------|
```

Type: 00 = no tape mounted
      01 = unlabelled
      10 = ANSI
      11 = IBM
L: 1 if file has lockword.
T: 1 if device is a tape drive.
B: 1 if tape is from Burroughs, which has incorrect block/record size in the HDR2 label. Code can be patched to correct the size.
HP: 1 if tape is Hewlett-Packard ANSI format.

VCB address: Pointer to VCB entry describing volume mounted on tape drive, only if linked. Otherwise, 0.
The VCB format is:

```
 0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
-----------------------------------------------------------------
| A | F | D |    Position    | W | SeqTyp| LblTyp| L | M | R | B |  0
|---------------------------------------------------------------|
|                          LDEV #                               |  1
|---------------------------------------------------------------|
|                           PIN                                 |  2
|---------------------------------------------------------------|
|                  File number (AFT index)                      |  3
|---------------------------------------------------------------|
|                   File sequence number                        |  4
|---------------------------------------------------------------|
| S | R | D | C |  Density   | V |         Reel number          |  5
|---------------------------------------------------------------|
|                     Expiration date                           |  6
|---------------------------------------------------------------|
|                                                               |  7
|                        File name                              |  10
|

|                                                               |
|                                                               |  16
|                               +-------------------------------|
|                               |                               |  17
+---------------------------------------------------------------+
|                                                               |  20
|                                                               |  21
|                         Lockword                              |  22
|                                                               |  23
|---------------------------------------------------------------|
|                                                               |  24
|                   Volume set identifier                       |  25
|
```

VCB (Cont.)

```
|                                                            | 26
|------------------------------------------------------------|
|                                                            | 27
|                                                            |
|                         Volume name                        | 30
|                                                            |
|                                                            | 31
|------------------------------------------------------------|
```

A: ASCII FOPTION
F: Flush bit - operator did REPLY <pin>,0.
D: DEVREC Wait (used with reelswitching).
Position: Gives head position within logical file.
0 = at load point (LDPNT)
1 = HDR1 label next (H1NX)
3 = after HDR2 label (AH2)
4 = after user header labels (AHU)
6 = data next (DNX)
7 = after data (AD)
8 = EOF1/EOV1 label next (T1NX)
10 = after EOF2/EOV2 label (AT2)
11 = after user trailer labels (ATU)
W: Write access specified.
SeqTyp: File open sequencing type.
0 = match filename
1 = NEXT
2 = ADDF
3 = use file sequence number
LblTyp: As in LCB entry.
L: Linkwait - mark left by CREATETLTENT for LINKLABEL.
M: Mount wait - waiting for operator to mount tape on FOPEN.
R: Reelswitch wait - waiting for next reel.
B: Busy bit - this entry is in use.

LDEV #: Logical device number of tape drive with this volume, only if linked. Otherwise, 0.

S: STORE tape.
R: REELSWITCH has been done. Used by STORE/RESTORE to handle STORE label and directory file.
D: Next file is directory. Used by STORE.
C: VOL1 label is to be created (written).
Density: volume set density. During a volume set open, contains the density requested by the user in FOPEN. Once the volume set is open, contains the actual density of the volume set. Only valid for tapes on variable density tape drives.
0 = default density for volume set open
1 = 1600 BPI
2 = 6250 BPI
V: 1 if volume set is being opened. Reset after completion of FOPEN.

Volume Recognition

Volume recognition is the responsibility of DEVREC, which reads the first record of a newly-mounted tape on an unowned drive and passes the record to AVREC. AVREC may see: VOL1 in the first 4 bytes, in ASCII, in which case the tape is ANSI; VOL1 in the first 4 bytes, in EBCDIC, in which case the tape is IBM; Anything else, in which case the tape is considered unlabelled.

If the tape is unlabelled, AVREC reports to DEVREC that no further action is required. If the tape is labelled, AVREC wants to see the first HDR1 label, so asks DEVREC to read another record. (Unfortunately, DEVREC cannot be stopped long enough for AVREC to do its own read.) When the HDR1 record is found, the volume entries can be searched to see if there is a pending request for this volume. If so, the waiting process is restarted.

If the system has been restarted with tapes mounted, there will not be interrupts to alert DEVREC. The procedure RECOGNIZE is called when needed to see if any such tapes exist.

Opening a File

FOPEN gets into the tape label code in three different places. The first is to call CREATETLTENT, which parses the string passed in the FORMSMSG parameter to identify the labeled tape file required. If there is no existing corresponding entry in the volume area, this is a volume set open, and a new volume entry is created. There may be an existing entry (if the tape was FOPENed and FCLOSEd with disposition 2 or 3), in which case there is an associated LDEV entry for the drive on which the tape was left mounted by the prior operation; in this case, the new information is stuffed into the existing volume entry. A bit (LINKWAIT) is left set to mark the entry for LINKLABEL.

The second entry is through LINKLABEL, which is called from ALLOCATE. At this time, it is necessary to identify the LDEV to be used for the tape. If no LDEV is associated, the LDEV entries are searched to see if the operator has already mounted the required tape; if so, the volume and LDEV entries are cross-tied and LINKLABEL is done. If the search turns up nothing suitable, the operator is requested to mount the appropriate tape, and the procedure waits for either a REPLY or for AVREC to discover the appearance of a suitable tape and restart the process. If the operator enters a reply, it is validated.

The third entry is through POSITION, which is responsible for positioning the tape to the requested file. At the file, the HDR1 and HDR2 label are examined as required to determine the file characteristics.

Reading and Writing Files

All procedures which move tape go through the catchall procedure CHECKUL, which takes care of necessary labeled tape doings. The code insures that the sequence: header labels (including user labels), data, trailer labels

(including user labels) is maintained. There is a separate CASE leg for each such procedure.

If an EOT reflective mark or an EOF in data is found, REELSWITCH is called (principally from the file system procedure IOMOVE) to call for the next reel, if any. If another reel is needed, the tape drive is set Unowned so that AVREC will be called to recognize the new tape when it is mounted. REELSWITCH returns to its caller when it is satisfied that an appropriate tape is mounted.

Closing Files

FCLOSE calls CHECKUL to handle writing EOF1 and EOF2 if needed and resolving the tape position. If the disposition is 3, the tape is left positioned at the next file. If the disposition is 2, the tape is supposed to be left at the beginning of the current file, but the code does not presently provide for reelswitching if the present file began on a prior reel.

At present, ensuing volumes of a multi-volume set must be mounted on the same drive as the first, mostly because neither the file system nor STORE-RESTORE was capable of dealing with LDEV changes in the middle of a file. REELSWITCH reports the LDEV being used, however, so that the capability of using a different LDEV can be added in the future.

Store-Restore

Complications ensue on labeled STORE-RESTORE tapes because there needs to be a file directory at or near the beginning of each tape of a multi-volume set; RESTORE uses this directory to determine whether the specified File(s) can exist on this tape. Because the reel switching process would otherwise be invisible to STORE-RESTORE, special bits (VCB'RSWDONE and VCB'WRITDIR) are kept to enable special intrinsics callable by STORE-RESTORE to report whether a directory needs to be written or is about to be encountered.

The special procedure NEXTTAPEFILE is used by STORE-RESTORE in lieu of doing a FCLOSE(,3) followed by an FOPEN to get to the next file. This permits cleaner handling of both REPLY 0 and Forward Space (logical) File over a Reelswitch, as well as saving the time needed to tear down and reconstruct all the control blocks.

Miscellaneous

PVOLID is used by the SHOWDEV command processor (in SPOOLCOMS) to obtain the name of the volume on the specified drive without having to know the structure of the tape label table. For the same reason, TGETINFO is used by the FFILEINFO intrinsic (in FILEIO) to get labeled tape information.

System Failure 86 in MPE is defined as a major problem in LABSEG. Generally speaking it is a problem with the TLT setup, for example if LABSEG cannot find an LDEV in the table.

Breakpoint Table

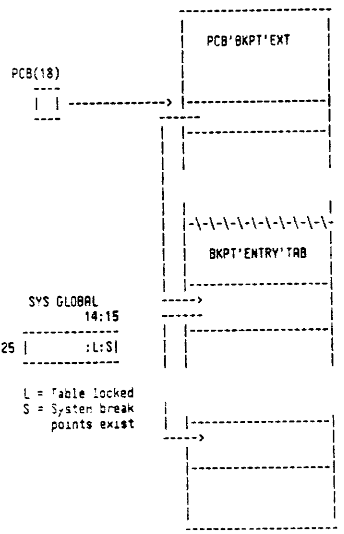
DST = 30(10) = %36

The break point table is divided into 2 sections:

1) PCB BREAKPOINT EXTENSION TABLE (PCB'BKPT'EXT)
This table contains the heads of the breakpoint chains

2) BREAKPOINT ENTRY TABLE (BKPT'ENTRY'TAB)
This table contains the actual entries

General Layout



PCB Breakpoint Extension Table

```
|    # ENTRIES        |     ENTRY SIZE =   1
-----------------------
| HEAD SYSTEM LIST    |     FREE ENTRY =   0
-----------------------
| # USED USER ENTRIES |     ACTIVE ENTRY = Index 1st Entry
-----------------------                    in breakpoint
|                     |                    chain
     USER ENTRIES

|                     |
|                     |
-----------------------
```

Breakpoint Entry Table

```
            ENTRY (0)                  FREE ENTRY
      -----------------------     -----------------------
  0   |# WORDS BREAKPOINT TAB |    |1:      SIZE           |
      -----------------------     -----------------------
  1   |   HEAD FREE LIST     |    |   FORWARD LINK        |
      -----------------------     -----------------------
  2   |     # WORD USED      |    |   BACKWARD LINK       |
      |----------------------|    -----------------------
  3   |   MAX # WORD USED    |    |                       |
      |----------------------|    |                       |
 4-6  |       UNUSED         |
      -----------------------
                                  |                       |
           LAST ENTRY             |                       |
      -----------------------     -----------------------
  0   |                    1|
      -----------------------
```

The breakpoint entry table consists of variable length entries
The mininum entry size is 7.

Active Entry

```
        0|1:2:3|4:5:6|7:8:9|0:1:2|3:4:5
        -------------------------------
       |0|P:L:V|D:F:T|U:P:C|U: SIZE     |
   0   | | : : | : : | :M: |P:          |
       ---------------------------------
   1   |M|           UNUSED             |
       ---------------------------------
   2   |          BLOCKLABEL            |
   3   |                                |
       ---------------------------------
   4   |             PLOC               |
       ---------------------------------
   5   |          INSTRUCTION           |
       ---------------------------------
   6   |             LINK               |
       ---------------------------------
       |          USERLABEL             |.
       ---------------------------------  .
       |                                |  .
       |                                |   .
            CONDITION/COUNT                   variable
                                             .
       |                                |   .
       ---------------------------------   .
       |  COND DESCRIPTOR               | .
       ---------------------------------
```

Breakpoint Entry Table (Cont.)

```
ENTRY(0).(0:1) = FR:    FREE ENTRY
                          1 = FREE
                          0 = USED
ENTRY(0).(1:1) = P:     PRIVILEGED MODE BREAKPOINT
                          1 = PRIV.
                          0 = NON-PRIV
ENTRY(0).(2:1) = L:     PROCESS-LOCAL BREAKPOINT
                          1 = PROCESS-LOCAL
                          0 = SYSTEM
ENTRY(0).(3:1) = V:     VALIDATION BIT
                          1 = INSTRUCTION IN ENTRY(3)
                          0 = INSTRUCTION NOT IN TAB.
ENTRY(0).(4:1) = D:     DOUBLE TRAP
                          1 = BREAKPOINT OSCILLATES BETWEEN
                              P/P+1
                          0 = NOT DOUBLE TRAP
ENTRY(0).(5:1) = F:     FAKE 'DUMMY' TRAP
                          1 = BREAKPOINT AT P+1
                          0 = BREAKPOINT AT P (ORIG. LOC)
ENTRY(0).(6:1) = T:     TWO WORD INSTRUCTION
                          1 = TWO WORD INSTRUCTION
                          0 = NOT TWO WORD INSTRUCTION
ENTRY(0).(7:1) = U:     USER LABEL PRESENT
                          1 = TRAP TO USER SUPPLIED LABEL
                          0 = TRAP TO DEBUG
ENTRY(0).(8:1) = PM:    PERMANENT BREAKPOINT
                          1 = PERM
                          0 = TEMPORARY
ENTRY(0).(9:1) = C:     CONDITION/COUNT
                          1 = CONDITION/COUNT SPECIFIED
                          0 = NO COND/COUNT
ENTRY(0).(10:1) = UP:   UPDATING
                          1 = ENTRY IN PROCESS OF BEING
                              UPDATED/REMOVED
                          0 = NOT BEING UPDATED/REMOVED
ENTRY(1).(0:1) = M      USER PLABEL MODE
ENTRY(6) = LINK:        LINK
                          0 = END OF CHAIN
                          >0= INDEX NEXT ENTRY
```

Breakpoint Entry Table (Cont.)

```
           COUNT                        CONDITION
      -----------------           -------------------
   1) | ORIGINAL CNT.  |       2) | OPERAND1          |
      -----------------           -------------------
      | # OF HITS      |          | OPERAND2          |
      -----------------           -------------------
      |             1  |          |OPT1|OPt2| RELOP   |
      -----------------           -------------------

RELOP -> (8:8) RELOP NUMBER:
               3  = LT      9  = LTE
               4  = GT      10 = GTE
               5  = EQ      11 = NEQ
OPT1  -> (0:2) OPERAND1'S TYPE
OPT2  -> (2:2) OPERAND2'S TYPE

OPERAND TYPES:
   0 ->  CONSTANT (SINGLE WORD)
   1 ->  ADDRESS (DOUBLE WORD)
   3 ->  INDIRECT ADDRESS (TRIPLE WORD)

OPERAND FORMS:
   CONSTANT ->  ------------
                | CONST    |
                ------------

   ADDRESS  ->  ------------
                | REG | BASE|
                ------------
                |  OFFSET  |
                ------------
                |IND. OFFSET|   (TYPE 3 ONLY)
                ------------

   REG      ->  (0:6) CORRESPONDING INDEX INTO 'REGY':
                      3  = A      10 = DL
                      4  = SY     11 = Q
                      7  = DA     12 = S
                      8  = DX     17 = EA
                      9  = DB
   BASE     ->  (6:10) SEG #/BANK #
```

Timer Request List (TRL)

The system clock interrupts every 100 ms, with the CR being automatically cleared. An exception is the Shared Clock Interface measurement service which allows rates as fast as 5 ms. The interrupt handler is the procedure TICK. On entry, DB is pointing to the base of timer request list. Besides timeout requests, the clock also controls time slicing.

```
        --------------------------------|
    / 0|      NUMBER OF ENTRIES         |
    |  |--------------------------------|
    | 1|       ENTRY SIZE (4)           |
ENT0|  |--------------------------------|
    | 2|       FREE LIST PTR            |
    |  |--------------------------------|
    \ 3| # of days since last start     |<- HP-IB Systems only
       |--------------------------------|
    / 4|       QUANTUM/100 ms           |  QTIME
    |  |--------------------------------|
    | 5|                                |
    |  |        TIME OF DAY*            |  DTIME*
ENT1| 6|                                |
    |  |--------------------------------|
    \ 7|    YEAR       |  JULIAN DAY    |
       |--------------------------------|
    / 8| PTR TO MOST ACTIVE REQUEST     |  HEAD
    |  |--------------------------------|
    | 9|        TRACE WORD              |
ENT2|  |--------------------------------|
    |10|             0                  |
    |  |                                |  dummy time
    \11|             0                  |  ----------
       |--------------------------------|       |
    /12|A|   CODE     |  INDEX OF NEXT  |       |
    |  |--------------------------------|       |
    |13|            REQ                 |       |
ENT3|  |--------------------------------|  assignable
    |  |    TIME TO SERVICE AFTER       |  entries
    \  | REQUEST IN FRONT (UNIT= 100ms)|       |
       |--------------------------------|       |
       |                                |       |
       |                                |       |
     A:  0 if inactive request              ----------
         1 if active request
```

TRL (Cont.)

CODE & REQ indicate the type of request.

CODE:	REQ:	TYPE:
0	DITP	Hangup
1	DITP	Carrier failure
2	DITP	202 turnaround
3	DITP	Read
4	DITP	Logon
5	PCBB index to process	Delay
6	DITP	LP not ready
7	DITP	2640
%10	Port mask	Msg port timeout
%11	DITP	Block mode read timeout (30 secs)
%12	PCBB index to process	Watchdog timer for process

The list of pending requests is kept ordered by time with later entries at the tail.

%20-%37	DITP	SIO device timeout: DIT8. (code_1 on expiration, cleared on Timereq.
%5/%6	*DTIME	For Series 30/33, DTIME is # of TICS (0.091457 ms) since last midnight.

MPE User Logging

MPE USER LOGGING enables users and subsystems to log changes to data sets on disc or serial files. This "change" file can later be used to recover data lost due to a system or program failure. The log file can itself be used for auditing purposes.

General Design Overview

Hardware Environment

No special hardware is required to operate the system. However, if logging to a tape file is desired, the hardware configuration must include a tape drive. If there is no tape drive, then may log to a serial disc class device.

Software Environment

MPE User Logging is an integral part of MPE. No other special software is required.

Design Narrative

User Logging enables users and subsystems to journalise additions and modifications to MPE and subsystem files. The journal can reside on either disc or serial logfiles.

User Logging consists of a logging process, a memory buffer, a disc resident logging buffer (for serial logging) and a user defined destination log file on disc or serial media.

The logging process has two functions depending on whether the destination file resides on disc or serial media. If the destination file is serial, the logging process performs all output to the destination file. If the destination file is on disc, the logging process allocates additional space (extents) as it is required by the user.

The logging buffer is divided into communication and buffer areas. The communication area is used to pass information among the users and the logging process. This information includes status of the logging process and logging file, space remaining in the logging file and error information important to users or the logging process. The buffer portion of the logging data segment blocks inputs into the logging file before the data is actually posted. The buffer is flushed any time a user requests to close a log file or when a logging process is terminated. (The buffer is also flushed by the begin/end transaction or buffer flush requests).

Error Recovery Description

The error recovery mechanisms provided by User Logging are: power fail recovery and recovery from system failure.

Power failure recovery applies only to tape log files since MPE provides adequate recovery for disc files during power fail. When a power failure is detected, a message will be printed on the console asking the operator to place the tape drive back on-line. (If the operator places the tape on-line before the message valid data may be overwritten). (To reset the tape drive the operator must hit the load button until the tension returns to the drive. Then hit the reset button followed by placing the tape drive back on-line). At this time the log process will recover the file by rewinding to the load point and then forward spacing to the point where the power fail occurred. Writing to the log file will continue at that point.

In the event of a system failure, the warm start load option initiates recovery of User Logging files. In the case of a serial file, the file is read and compared to the disc logging buffer. All records found in the disc buffer that are not on the serial log file are posted and a proper end of file written. If the destination file is a disc file, all records are read and verified and an end of file posted to the file. In order to continue logging to a User Logging file that has been recovered in this manner, the logging process for the file must be restarted using the console command :LOG.

NOTE:

> Any records in the buffer area of the logging buffer will be lost.
>
> User logging has been enhanced to work with labeled serial discs. Internally the log process handles serial disc (or cartridge tape) log files the same as for tape files.

Design Structures

User Logging Table

ENTRY SIZE = #38 words
DST %33

Table containing an entry for each activated user logging process. Each entry is created when the process is started, and deleted when the process terminates. (Via :LOG command). The information is extracted from the Logging Identifier Table (LIDTAB).

#	ENTRY 0	%
0	NUMBER OF ENTRIES	0
1	FREE ENTRY HEAD PT.	1
2	INUSE ENTRY HEAD PT.	2
3	NEXT BUFFER NUMBER	3
4	MAX # PROCESSES	4
5	MAX # USERS/PROCESS	5
6		6
7	ENTRY SIZE	7
37	. . .	45

WORD ENTRIES

NUMENTRIES	=	LOGTAB
FREE	=	LOGTAB(1)
INUSE	=	LOGTAB(2)
BUFNUM	=	LOGTAB(3)
MAXLOGPROC	=	LOGTAB(4)
MAX'USR'PROC	=	LOGTAB(5)
LOGTAB'ESIZE	=	LOGTAB(7)

NUMENTRIES
The number of entries in the logging table.

FREE
A table relative pointer to the first free entry in the logging table. (-1 = table full).

INUSE
A table relative pointer to the first entry in the logging table that is being used (-1 = no entries in use).

BUFNUM
The number of the buffer associated with this logging process. Used to create the name of buffer file if serial logfile. (i.e. ULOGxxxx.PUB.SYS).

MAXLOGPROC
The maximum number of user logging processes allowed.

MAX'USR'PROC
The maximum number of users per logging process.

LOGTAB'ESIZE
The size (in words) of each entry in the table.

Typical Entry

#		%
0	LOGGING IDENTIFIER	0
4	BUFFER NAME	4
8	FILE NAME	10
12	LOCK WORD	14
16	GROUP	20
20	ACCT	24
24	NUMBER OF USERS	30
25	BUFFER DST NO	31
26	LOG STATUS	32

27	CURR AUTO | CURR TYPE	33
28	LOG DEV	34
29	LOG PCB #	35
30	SWITCH FLAG	36
31	NEW AUTO | NEW TYPE	37
32	ADDRESS OF LOGGING BUFFER	40
34	SIZE OF LOGGING BUFFER	42
36	FWRD ENTRY PT	44
37	BWRD ENTRY PT	45

TABINDEX	=	WORD INDEX TO CURRENT ENTRY
BTABINDEX	=	BYTE INDEX TO CURRENT ENTRY
DTABINDEX	=	DOUBLE INDEX TO CURRENT ENTRY
LGNAME	=	BTABINDEX
BNAME	=	BTABINDEX+8
LFNAME	=	BTABINDEX+16
LFLOCKW	=	BTABINDEX+24
LFGROUP	=	BTABINDEX+32
LFACCT	=	BTABINDEX+40
NUMUSERS	=	TABINDEX+24
DST	=	TABINDEX+25
STATUS	=	TABINDEX+26
LGAUTO	=	TABINDEX+27.(0:8)
LGTYPE	=	TABINDEX+27.(8:8)
LGDEV	=	TABINDEX+28
PIN	=	TABINDEX+29
LGSWITCH	=	TABINDEX+30
LGNEWAUTO	=	TABINDEX+31.(0:8)
LGNEWTYPE	=	TABINDEX+31.(8:8)
LGADDR	=	DTABINDEX+16
BSIZE	=	DTABINDEX+17
NEXT	=	TABINDEX+36
PREV	=	TABINDEX+37

LGNAME
The name of the logging process (logging identifier).

BNAME
The name of the disc buffer used if the logging process destination file is a serial file. This is a file that resides in PUB.SYS. The format of the name is ULOGxxxx where xxxx is the buffer number padded on the left with zeros.

If the switch fl.g is true, the following will be the fully qualified file name of the new log file.

LFNAME
The name of the logging file.

LFLOCKW
The lockword of the disc logging file.

LFGROUP
The group that the destination logging file resides in if the file is a disc file.

LFACCT
The account that the destination logging file resides in if the file is a disc file.

NUMUSERS
The number of users currently accessing the logging file.

DST
The dst number of the logging data segment (LOGBUFF). (-1 = LOGBUFF not created yet)

STATUS
The status of the logging process.

INITIALIZING	=	-1
INACT	=	0
ACT	=	1
RECOVERING	=	2

LGAUTO
True if the automatic changelog facility was enabled. (Not used - for future use).

LGTYPE
The type of destination file of the logging process.

DISC	=	0
TAPE	=	1
SDISC	=	2
CTAPE	=	3

LGDEV
The logical device number of the disc logging file or the disc logging buffer.

PIN
The PCB number for the logging process (PIN * PCBSIZE).

LGSWITCH
Flag indicating a CHANGELOG is pending (if true). (Not used - for future use).

LGNEWAUTO
True if the automatic changelog facility was requested for the new log file. (Not used - for future use).

LGNEWTYPE
If a switch is pending, this will be the type of the new log process. (-1 = no switch pending). (Not used - for future use).

LGADDR
Sector number of the current extent in the disc logging file or the disc buffer file. (Disc buffer file has only 1 extent)

BSIZE
The number of records in the current extent (for disc logging) or the number available in the disc logging buffer.

NEXT
A table relative pointer to the next entry in the logging table. (-1 = this is last entry)

PREV
A table relative pointer to the previous entry in the logging table. (-1 = this is first entry)

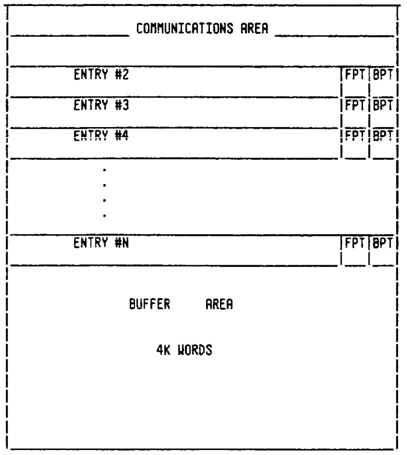
User Logging Buffer

There will be one of these tables around for the life of any active user logging process. The table consists of three parts:

COMMUNICATIONS AREA - Information about status of the process, etc. that is common to all users of the process. Also the cells for messages to/from the process.

USER ENTRIES - Information for a specific user of the process. One of these for every user of a process (Setup by OPENLOG, released by CLOSELOG).

BUFFER AREA - Buffer used to hold logging records from all users before writing to the log file.



COMMUNICATIONS AREA

#		%
0	LOGGING IDENTIFIER	0
4	SWITCH FLAG	4
5	NEW AUTO | NEW TYPE	5
6	AUTO | TYPE	6
7	BUFFER DST	7
8	LOG PIN	10
9	NUMBER OF USERS	11
10	MAX NUMBER OF USERS	12
11	NEXT USER NUMBER	13
12	SLEEP COUNT	14
13	STATE	15
14	MSG	16
15	LOG MSG	17
16	USER MSG	20
17	LOG ERROR	21
18	LOG DEVICE	22
19	BUFFER SPACE	23
20	USED SPACE IN BUFFER	24
21	FILE SET NUMBER	25
22	LOG ADDRESS	26
24	INPUT RECORD	30
26	FILE	32

	SIZE	
28	FILE SPACE	34
30	TOTAL RECORDS	36
32	MAX SIZE	40
34	LAST EXTENT	42
35	EXTENT	43
36	RESOURCE	44
40		50
48	IN USE HEAD PTR	60
49	FREE HEAD PTR	61

LOGID	=	BLOGBUFF(0)
SWITCH'	=	LOGBUFF(4)
NEWAUTO	=	LOGBUFF(5).(0:8)
NEWTYPE	=	LOGBUFF(5).(8:8)
AUTO	=	LOGBUFF(6).(0:8)
LOGTYPE	=	LOGBUFF(6).(8:8)
BDST	=	LOGBUFF(7)
LOGPIN	=	LOGBUFF(8)
NUMUSER	=	LOGBUFF(9)
MAXUSER'	=	LOGBUFF(10)
USERNO	=	LOGBUFF(11)
SLPCT	=	LOGBUFF(12)
STATE	=	LOGBUFF(13)
MSG	=	LOGBUFF(14)
LOGMSG	=	LOGBUFF(15)
USERMSG	=	LOGBUFF(16)
LOGERR	=	LOGBUFF(17)
LOGDEV	=	LOGBUFF(18)
BSPACE	=	LOGBUFF(19)
BUFUSED	=	LOGBUFF(20)
VSETNO	=	LOGBUFF(21)
LOGADDR	=	DLOGBUFF(11)
INBUFREC	=	DLOGBUFF(12)
FSIZE	=	DLOGBUFF(13)
FSPACE'	=	DLOGBUFF(14)
TRECS	=	DLOGBUFF(15)
MAXFSPACE	=	DLOGBUFF(16)
LASTEXT'	=	LOGBUFF(34)
EXTENT	=	LOGBUFF(35)
RESOURCE	=	DLOGBUFF(18)
UHEAD	=	LOGBUFF(48)
FHEAD	=	LOGBUFF(49)

LOGID
The name of the logging process.

SWITCH'
True if log file switch is pending. (Not used - for future use).

NEWAUTO
True if the automatic changelog option has been specified for the new log file. (Not used - for future use).

NEWTYPE
If a switch was requested, this will be the type of the new logging file. (-1 = no switch pending) (Not used - for future use).

AUTO
True if the automatic changelog option was specified for the current log file. (Not used - for future use).

LOGTYPE
The type of destination file for the logging process.

```
     DISC  = 0
     TAPE  = 1
     SDISC = 2
     CTAPE = 3
```

BDST
The data segment number of this table.

LOGPIN
This is the PCB number for the logging process (PIN*PCBSIZE).

NUMUSER
The number of users currently accessing the logging file.

MAXUSER'
The maximum number of users allowed to access the logging file.

USERNO
The next sequential number to be assigned users accessing the system. It will get incremented for every unique OPENLOG - used as the log # in the logging record format.

SLPCT
The number of users currently waiting for activation by the logging process.

STATE
The state of the user logging process.

```
     INACTIVE  = 0
     ACTIVE    = 1
```

MSG
An internal message word used to indicate an error or operator request.

```
     6 - Continue processing, all is fine.
     2 - Suspend - error reading buffer file or writing to serial file
     3 - Stop - set when issue :LOG logid,STOP  or when an EOF condition is
     found on the disc log file.
```

LOGMSG
A messages from the logging process.

```
   6 - Continue processing, all is fine.
  15 - EOF - if there are no more extents available to be
       allocated.
  12 - Disc space - could not allocate the new extent because
       no space left in the group.
   9 - Write error - error occurred while writing to log file
```

USERMSG
A messages from the user process.

```
   6 - Continue processing, all is fine.
  12 - Disc space - user process needs another extent allocated
       for disc logging.
```

LOGERR
Last error found. After changelog:

```
  +N - File System error number encountered
   0 - No error
  -1 - New disc log file was not empty
  -2 - New disc log file did not have file code LOG
  -3 - New disc file is too small
```

(Not used - for future use).

LOGDEV
The logical device number of the current extent of the disc log file or the disc buffer file (buffer file has only 1 extent).

BSPACE
The amount of space, in records, that are currently available to the users. On the last block of the last extent, one record will be saved by the logging process so that the proper close information can be posted to the file - either the trailer record (if the log logging process is stopped) or the change'to'new record because of an EOF condition (and the AUTO option had been specified).

BUFUSED
The number of records currently in the buffer. On all extents, except the last extent BUFSPACE+BUFUSED = 32 (number of records in a complete block). However, on the last block of the last extent this will NOT be true since one record is always held in reserve by the logging process.

VSETNO
This shows the order in the log file "set" of the currently opened log file. (Not used - for future use).

LOGADDR
The disc address of the current extent of the disc log file. If it's a serial file, this is the disc address of the disc buffer for the file.

INBUFREC
The record number of the next block to be written to the logging destination file or the disc logging buffer for serial files. (Used as an offset into the current extent for the writes - since each record is one sector in length).

FSIZE
The current extent size of the logging destination file or disc logging buffer file for serial destination files. (on the last extent this will be the last extent size minus 1).

FSPACE'
The space in records that remains in the current extent of the disc logging destination file or disc buffer for tape destination files. (On the last extent of the disc log file, this is the amount of space minus 1).

TRECS
The total number of records written to the logging destination file (including those records currently in the buffer).

MAXFSPACE
The total file size, in records, minus 1. (Need that last record to post close information).

LASTEXT'
The extent number of the final extent in the disc logging file or disc buffer file.

EXTENT
The current extent number of the disc logging file or disc logging buffer.

RESOURCE
Used for resource management (i.e. locking the LOGBUFF). Format is:
    RESOURCE + 0 = Owner PCB number
    RESOURCE + 1 = Head of impeded queue PCB number
    RESOURCE + 2 = Tail of impeded queue PCB number
    RESOURCE + 3 = Queue length

UHEAD
A table relative pointer to the first entry into the logging data segment. (-1 = no entries currently in use)

FHEAD
A table relative pointer to the first free entry in the logging data segment. (-1 = no free entries)

TYPICAL LOGBUFF ENTRY

#		%
0	USER NAME	0
4	GROUP NAME	4
8	ACCOUNT NAME	10
12	USER PCB #	14
13	OPENLOG COUNT	15
14	WAIT STATE	16
15	ERROR CODE	17
16	LOG NUMBER	20
17	SUBSYSTEM CODE	21
18	TOTAL RECORDS	22
23	FRWD ENTRY PTR	27
24	BKWRD ENTRY PTR	30

```
BINDEX    =    BYTE INDEX TO CURRENT ENTRY
INDEX     =    WORD INDEX TO CURRENT ENTRY
DINDEX    =    DOUBLE INDEX TO CURRENT ENTRY

USER      =    BINDEX
GROUP     =    BINDEX+8
ACCT      =    BINDEX+16

UPIN      =    INDEX+12
OPENCNT   =    INDEX+13
WSTATE    =    INDEX+14
ERROR     =    INDEX+15
LGNUM     =    INDEX+16
SCODE     =    INDEX+17

RECS      =    DINDEX+9

NENTRY    =    INDEX+23
PENTRY    =    INDEX+24
```

USER
The name of the user who opened the logging file through this entry.

GROUP
The group of the user who opened the logging file.

ACCT
The account of the user who opened the logging file.

UPIN
The PCB number of the user process (PIN * PCBSIZE).

OPENCNT
Counter of how many times this user called OPENLOG. (Incremented for every OPENLOG, decremented for every CLOSELOG). (Not used - for future use).

WSTATE
The wait status of the users process.
    INACTIVE = 0
    ACTIVE   = 1

ERROR
Used to hold error information for this user.
    -1 = No room in disc (or disc buffer) and NOWAIT.
    0 = O.K.

LGNUM
The logging number assigned to the user. (From USERNO in global area to be used as log # in the log record).

SCODE
The subsystem code for the caller. This applies only to privileged callers.

RECS
The number of records written by this user.

NENTRY
A table relative pointer to the next entry in the logging data segment. (-1 = this is the last entry)

PENTRY
A table relative pointer to the previous entry in the logging data segment. (-1 = this is the first entry)

User Logging Identifier Table

ENTRY SIZE = #33 words
DST %41

Table containing an entry for each potential logging process. Entries are added via :GETLOG and released via :RELLOG.

Entry #0

```
     #                          %

     0   ____________________   0
        |                    |
     1  |MAX NUMBER OF ENTRIES|  1
        |____________________|
     2  |                    |  2
        |____________________|
     3  |                    |  3
        |____________________|
     4  |     ENTRY SIZE     |  4
        |____________________|
        |         .          |
        |         .          |
    32  |         .          |  40
        |____________________|
```

ENTRIES

```
MENTRIES     =    LIDTAB(1)
ENTRYSIZE    =    LIDTAB(4)
```

MENTRIES
The maximum number of entries in the table. (i.e. maximum number of user logging processes. 1 entry for every process - activated or not).

ENTRYSIZE
The size of each entry in the table.

Typical Entry

```
   #   ____________________   %
   0  |                    |  0
      |-                  -|
      |     LOGGING        |
      |-                  -|
      |    IDENTIFIER      |
      |-                  -|
      |____________________|
   4  |                    |  4
      |-                  -|
      |     PASSWORD       |
      |-                  -|
      |                    |
      |____________________|
   8  |                    |  10
      |-     FILE         -|
      |                    |
      |-     NAME         -|
      |                    |
      |____________________|
  12  |                    |  14
      |-     FILE         -|
      |                    |
      |-   LOCK WORD      -|
      |                    |
      |____________________|
  16  |                    |  20
      |-     FILE         -|
      |                    |
      |-     GROUP        -|
      |                    |
      |____________________|
  20  |                    |  24
      |-     FILE         -|
      |                    |
      |-    ACCOUNT       -|
      |                    |
      |____________________|
  24  |                    |  30
```

Typical Entry (Cont.)

```
      |-                  -|
      |     USER'S         |
      |-                  -|
      |      NAME          |
      |-                  -|
      |____________________|
  28  |                    |  34
      |-    USER'S        -|
      |                    |
      |-    ACCOUNT       -|
      |                    |
      |____________________|
  32  |     LOG TYPE       |  40
      |____________________|
```

BYTE ENTRIES

```
LID      =    BLIDTAB
PW       =    BLIDTAB(8)
FNAME'   =    BLIDTAB(16)
LW       =    BLIDTAB(24)
FGROUP   =    BLIDTAB(32)
FACCT    =    BLIDTAB(40)
UNAME    =    BLIDTAB(48)
UACCT    =    BLIDTAB(56)
```

WORD ENTRIES

```
TYP      =    LIDTAB(32)
```

LID
The logging identifier name. This is a maximum of eight characters long.

PW
The pass word for the logging identifier. This is a maximum of eight characters long.

The following is the fully qualified file name of the current log file.

FNAME'
The name of the destination file.

LW
The lock word on the destination file if the file is on disc.

FGROUP

The group that the file resides in.

FACCT
The account that the destination file resides in.

UNAME
The name of the user who created the logging identifier.

UACCT
The account of the user who created the logging identifier.

TYP
The status of the entry.

```
-1 = null entry
 0 = disc logging file
 1 = tape logging file
 2 = serial disc logging file
 3 = cartridge tape logging file
```

Logging Record Format

RECORD SIZE = 128 words
USER AREA = 119 words

LOG RECORD AT OPENLOG

```
0    2    3    4    6    7     11   12         24   25    127
 ______________________________________________________________
|     |     |     |    |    |      |    |          |    |      |
| rec#|cksum|code |time|date| logid|log#|   creator|pcb |      |
|_____|_____|_____|____|____|______|____|__________|____|______|
```

USER OR SUBSYSTEM/CONTINUATION LOG RECORD (from WRITELOG)

```
0    2     3    4    6    7    8   9                        127
 ______________________________________________________________
|     |     |     |    |    |    |   |                         |
| rec#|cksum|code |time|date|log#|len|       user area         |
|_____|_____|_____|____|____|____|___|_________________________|
```

LOG RECORD AT CLOSELOG

```
0    2    3    4    6    7     11   12         24   25    127
 ______________________________________________________________
|     |     |     |    |    |      |    |          |    |      |
| rec#|cksum|code |time|date| logid|log#|   creator|pcb |      |
|_____|_____|_____|____|____|______|____|__________|____|______|
```

CRASH MARKER

```
0    2     3    4    6    7                                 127
 ______________________________________________________________
|     |     |     |    |    |                                  |
| rec#|cksum|code |time|date|                                  |
|_____|_____|_____|____|____|__________________________________|
```

HEADER RECORD (START/RESTART)

```
0    2    3    4    6    7       11                         127
 ______________________________________________________________
|     |     |     |    |    |      |                           |
| rec#|cksum|code |time|date| logid |                          |
|_____|_____|_____|____|____|______|___________________________|
```

TRAILER RECORD (STOP)

```
0    2    3    4    6    7       11                         127
 ______________________________________________________________
|     |     |     |    |    |      |                           |
| rec#|cksum|code |time|date| logid |                          |
|_____|_____|_____|____|____|______|___________________________|
```

NULL RECORD

```
0    2     3    4    6    7                                 127
 ______________________________________________________________
|     |     |     |    |    |                                  |
| rec#|cksum|code |time|date|                                  |
|_____|_____|_____|____|____|__________________________________|
```

BEGIN TRANSACTION MARKER

```
0    2     3    4    6    7    8   9                        127
 ______________________________________________________________
|     |     |     |    |    |    |   |                         |
| rec#|cksum|code |time|date|log#|len|       user area         |
|_____|_____|_____|____|____|____|___|_________________________|
```

END TRANSACTION MARKER

```
0    2     3    4    6    7    8   9                        127
 ______________________________________________________________
|     |     |     |    |    |    |   |                         |
| rec#|cksum|code |time|date|log#|len|       user area         |
|_____|_____|_____|____|____|____|___|_________________________|
```

CODE DEFINITION

```
  CODE.(8:8) =
         1   Open log record
         2   User/subsystem record (writelog)
         3   Close log record
         4   Header record
         5   Trailer record
         6   Restart record
         7   Continuation of a user or subsystem record
         9   Crash marker
        10   End transaction record
        11   Begin transaction record
     SPACE   NULL record
```

DATA FIELDS OF LOG RECORDS

REC#	=	DOUBLE INTEGER
CKSUM	=	INTEGER
CODE	=	INTEGER
TIME	=	DOUBLE (from intrinsic CLOCK)
DATE	=	INTEGER (from intrinsic CALENDAR)
LOGID	=	ASCII
LOG#	=	INTEGER
LEN	=	INTEGER
USERAREA	=	ASCII
CREATOR	=	ASCII
PCB	=	INTEGER

NOTE:

1. The checksum algorithm uses the exclusive or (XOR) function against a base of negative one.

2. Null record is used for filler.

3. The code word of the logging record can contain a subsystem code defined by the user in the first half of the word (0:8). User logging allows privileged users to pass this code in the index parameter of the Openlog intrinsic.

4. The "len" field will contain the entire length of the data in the transaction (i.e. the length passed to WRITELOG, BEGINLOG, ENDLOG). If a continuation record is part of the transaction, it will also contain the entire length of the data. For example, a length of 140 was passed to the intrinsic. The "len" field of the first record will be 140, the "len" field of its continuation record will also be 140 - even though the actual amount of data found in the first record will be 119 and the data found in the continuation record will be 21.
(Positive length = # words, negative length = # bytes)

```
          MEASINFOTAB                DST = 59 (% 73)
          -----------


        ------  ---------------------------------------
          |   0 | LDEV # OF MEASIO                     | MEASLDEV
          |     ---------------------------------------
          |   1 | MEASIO PLABEL                        | MEASPLAB
          |     ---------------------------------------
          |   2 | MEASIO DST #                         | MEASDSTN
          |     ---------------------------------------
 Reserved     3 |                                      |
 for MEASIO     ---------------------------------------
 control      4 |                                      |
          |     ---------------------------------------
          |   5 |                                      |
          |     ---------------------------------------
          |   6 |                                      |
          |     ---------------------------------------
          |   7 |                                      |
        -----   ---------------------------------------
          |  10 |                                      |
          |     ---------------------------------------
          |  11 |                                      |
          |     ---------------------------------------
          |  12 |                                      |
 Reserved       ---------------------------------------
 for         13 |                                      |
 performance    ---------------------------------------
 tuning        14 |                                      |
 parameters     ---------------------------------------
          |  15 |                                      |
          |     ---------------------------------------
          |  16 |                                      |
          |     ---------------------------------------
          |  17 |                                      |
        -----   ---------------------------------------
          |  20 | GLOBAL STATISTICS XDS NUMBER         | MEASSTATX-
          |     |                                      | DSNUM
          |     ---------------------------------------
          |  21 | PROCESS STATISTICS XDS BANK          | MEASPROC-
          |     |                                      | XDSBANK
          |     ---------------------------------------
          |  22 | PROCESS STATISTICS XDS BASE          | MEASPROC-
          |     |                                      | XDSBASE
          |     ---------------------------------------
          |  23 | PROCESS STATISTICS XDS NUMBER        | MEASPROC-
          |     |                                      | XDSNUM
          |     ---------------------------------------
          |  24 |    CLASS 14 STATISTICS XDS BANK      |
          |     ---------------------------------------
          |  25 |    CLASS 14 STATISTICS XDS BASE      |
          |     ---------------------------------------
```

```
|       ----------------------------------------
| 26 |    CLASS 14 STATISTICS XDS NUM.      |
|       ----------------------------------------
| 27 |    CLASS 13 STATISTICS XDS BANK      |
|       ----------------------------------------
| 30 |    CLASS 13 STATISTICS XDS BASE      |
|       ----------------------------------------
| 31 |    CLASS 13 STATISTICS XDS NUM.      |
|       ----------------------------------------
| 32 |    CLASS 12 STATISTICS XDS BANK      |
|       ----------------------------------------
| 33 |    CLASS 12 STATISTICS XDS BASE      |
|       ----------------------------------------
| 34 |    CLASS 12 STATISTICS XDS NUM.      |
|       ----------------------------------------
| 35 |    CLASS 11 STATISTICS XDS BANK      |
|       ----------------------------------------
| 36 |    CLASS 11 STATISTICS XDS BASE      |
|       ----------------------------------------
| 37 |    CLASS 11 STATISTICS XDS NUM.      |
|       ----------------------------------------
| 40 |    CLASS 10 STATISTICS XDS BANK      |
|       ----------------------------------------
| 41 |    CLASS 10 STATISTICS XDS BASE      |
|       ----------------------------------------
| 42 |    CLASS 10 STATISTICS XDS NUM.      |
|       ----------------------------------------
| 43 |    CLASS 09 STATISTICS XDS BANK      |
|       ----------------------------------------
| 44 |    CLASS 09 STATISTICS XDS BASE      |
|       ----------------------------------------
| 45 |    CLASS 09 STATISTICS XDS NUM.      |
|       ----------------------------------------
```

```
          |       ----------------------------------------
   reserved  .  |                                        |
        for       ----------------------------------------
 measurement .  |                                        |
   interface      ----------------------------------------
          |  .  |                                        |
          |       ----------------------------------------
          |  50 |CLASS 0 ENABLED   |CLASS 1 ENABLED      |
          |     |COUNT             |COUNT                |
          |       ----------------------------------------
          |  51 | CLASS 2 EN.CNT.  | CLASS 3 EN.CNT.     |
          |       ----------------------------------------
          |  52 | CLASS 4 EN.CNT.  | CLASS 5 EN.CNT.     |
          |       ----------------------------------------
          |  53 | CLASS 6 EN.CNT.  | CLASS 7 EN.CNT.     |
          |       ----------------------------------------
          |  54 | CLASS 8 EN.CNT.  | CLASS 9 EN.CNT.     |
          |       ----------------------------------------
          |  55 | CLASS 10 EN.CNT.| CLASS 11 EN.CNT.     |
          |       ----------------------------------------
          |  56 | CLASS 12 EN.CNT.| CLASS 13 EN.CNT.     |
          |       ----------------------------------------
          |  57 | CLASS 14 EN.CNT.| CLASS 15 EN.CNT.     |
       ----       ----------------------------------------
          |  60 |                                        |
          |       ----------------------------------------
          |  61 |                                        |
   reserved       ----------------------------------------
     for     62 |                                        |
   shared         ----------------------------------------
   clock     63 |                                        |
   interface      ----------------------------------------
   user      64 |                                        |
          |       ----------------------------------------
          |  65 |                                        |
          |       ----------------------------------------
          |  66 |                                        |
          |       ----------------------------------------
          |  67 |                                        |
       -----      ----------------------------------------
```

```
    -----  ----------------------------------------
        | 70 | M |          FLAG               | A |
        |      ----------------------------------------
   shared 71 |                XDSI                   |
             ----------------------------------------
   clock  72 |                XDS2                   |
             ----------------------------------------
interface 73 |                DCOUNT                 |
             ----------------------------------------
   cells  74 |                DLIMIT                 |
             ----------------------------------------
        | 75 |                TCOUNT                 |
        |    ----------------------------------------
        | 76 |                TLIMIT                 |
        |    ----------------------------------------
        | 77 |                DLABEL                 |
    -----    ----------------------------------------
        | 100 |  MONITOR BUFFER INDEX               | SMONIDX
        |    ----------------------------------------
        | 101 |  MEAS BUFFER                        | MEASBUFO
        |    ----------------------------------------
        | 102 |  MEAS BUFFER INDEX                  | MEASIDX
        |    ----------------------------------------
 reserved 103 |  MEAS ENABLED FLAGS                 | MEASMSKO
     for     ----------------------------------------
   event 104 |  MEAS ENABLED FLAGS                  | MEASMSK1
  logging    ----------------------------------------
         105 |  MEAS BUFFER BANK                    | MEASBUFBANK
        |    ----------------------------------------
        | 106 |                                     |
        |    ----------------------------------------
        |   . |                                     |
        |    ----------------------------------------
        |   . |                                     |
        |    ----------------------------------------
        | 116 |                                     |
        |    ----------------------------------------
        | 117 |                                     |
        |    ----------------------------------------
    -----
```

M: Interrupt has missed due to last interrupt handling.

A: Current interrupt handling active.

CHAPTER 18 MESSAGE FILES

Message File Data Structures

This chapter contains the data structures necessary to support message files. The first section details the message file's version of the familiar file system data structure; ie, the file label, file control block, access control block, etc..

The second section shows the tables used by the basic IPC mechanism which is a set of internal, MPE procedures designed to support the "boundary conditions" of IPC files. For example, signaling a no wait reader that its record has arrived. See the section's introduction for a detailed description.

File Structure

File Label/FCB Extent Map

```
 ___________________________  End of file block    Start of file block
| Disc addr of extent 0     |   .                    .
|---------------------------|   .                    .
| Disc addr of extent 1     |   v                    .
|---------------------------|   -                    .
| Disc addr of extent 2     |                        .
|---------------------------|                        .
| Disc addr of extent 3     |                        .
|---------------------------|                        .
\                           \                        .
|---------------------------|                        .
| Disc addr of extent n-1   |                        v
|---------------------------|                        -
| Disc addr of extent n     |
|---------------------------|
```

The EOF and SOF are examples only, meant to show:

1) The start of file moves into the extent map as records are read
2) The file can wrap around and, hence, cause the SOF to be greater than the EOF.

When a file becomes empty the SOF and EOF are reset to the first block of extent zero.

Each extent is composed of a number of blocks. Extents all have the same number of blocks. Extent zero also contains space for the file label and user labels in the exact same format as standard files. Starting with block zero, sufficient blocks are allocated to the file label/user labels to satisfy their space requirements.

Extents outside of the SOF/EOF range may not exist. They are deleted at close time when there are no more writers accessing the file.

Block Structure

```
|---------------------------|   **********************************
| First data record         |
|---------------------------|   Exact same format as standard
| Second data record        |   variable length blocks.
|---------------------------|
\                           \
|---------------------------|
| Last data record          |
|---------------------------|
| Record delimiter (-1)     |
|---------------------------|   **********************************
|                           |
| Empty space (next record  |
| would not fit)            |
|                           |
|---------------------------|
| Header delimiter (%77)    |
|---------------------------|
| Last header record        |
|---------------------------|
\                           \
|---------------------------|
| Second header record      |
|---------------------------|
| First header record       |
|---------------------------|
```

Separating the data portion of the records from their header enables the standard file system access procedures to read the records with no knowledge that they are msg file records.

Record Format

```
 ---------------------------
| Number of bytes in record |
|---------------------------|
| First data word of record |
|---------------------------|
\                           \
|---------------------------|
| Last data word of record  |
|---------------------------|
```

Length word's value does not include itself.

Header Format

```
|---------------------------|
| C|LC|       | Header Type| 0
|---------------------------|
| Writer's ID               | -1
|---------------------------|
```

C (0:1) - Set on if this was the last record written before
           the system crashed. This bit is set on by the
           first open on the file after the crash.

LC (1:1)- Valid only for close headers. Set to one if this is
           the last writer to close the file.

Type(8:8)- 0 data
           1 open
           2 close

Message Access Control Block

Notes:

1. Words/fields that do not pertain to message files are left blank.
2. This diagram shows the "combined" ACB as it appears to the message access procedures (the procedures in IPC). Thus it is a combination of the LACB and the PACB.

```
   |-----------------------------------------------|
-5 | DST number of the PACB                        |-5
   |-----------------------------------------------|
-4 | PACB control block vector table address       |-4
   |-----------------------------------------------|
-3 | DST number of the LACB                        |-3
   |-----------------------------------------------|
-2 |                                               |-2
   |-----------------------------------------------|
-1 |                                               |
   |-----------------------------------------------|
 0 |   | Size of the ACB including buffers (words) | 0
   |-----------------------------------------------|
 1 | File Number                                   | 1  *
   |-----------------------------------------------|
 2 | File name                                     | 2  *
   |------------------          -------------------|
   \                                               \  *
   |-----------------------------------------------|
 6 | Foptions                                      | 6  *
   |-----------------------------------------------|
 7 | Aoptions                                      | 7  *
   |-----------------------------------------------|
 8 | Record size (bytes)                           | 10 *
   |-----------------------------------------------|
```

```
 9 | Block size (words)                                | 11 *
   |---------------------------------------------------|
10 |                                                   | 12
   |---------------------------------------------------|
11 | Carriage control code (writers)                   | 13 *
   |---------------------------------------------------|
12 | No wait I/O target                                | 14 *
   |---------------------------------------------------|
13 | No wait I/O count                                 | 15
   |---------------------------------------------------|
14 | Error code                                        | 16 *
   |---------------------------------------------------|
15 | Transmission log (units same as last read/write)  | 17 *
   |---------------------------------------------------|
16 | Total number of unread records (includes opens    | 20
   |-------------                      ----------------|
17 | and closes)                                       | 21
   |---------------------------------------------------|
18 | Block number of the file's tail (relative to the  | 22
   |----------------                 ------------------|
19 | start of file block)                              | 23
   |---------------------------------------------------|
20 | Logical record transfer count                     | 24
   |----------------                 ------------------|
21 |                                                   | 25
   |---------------------------------------------------|
22 | Physical block transfer count                     | 26
   |----------------                 ------------------|
23 |                                                   | 27
   |---------------------------------------------------|
24 | DST REL ADDR of Read Header                       | 30
   |---------------------------------------------------|
25 | DST REL ADDR of Write header                      | 31
   |---------------------------------------------------|
26 | FCB DST                                           | 32
   |---------------------------------------------------|
27 | FCB vector table offset                           | 33
   |---------------------------------------------------|
28 | Share count ( number of LACBs )                   | 34
   |---------------------------------------------------|
29 | Access class, status, etc.                        | 35
   |---------------------------------------------------|
30 | Logical device number                             | 36
   |---------------------------------------------------|
31 |           |Wrt buf indx|             | # buf - 1  | 37
   |---------------------------------------------------|
32 | DST relative address of next read record          | 40
   |---------------------------------------------------|
33 | Size of the buffer (words)                        | 41
   |---------------------------------------------------|
34 | Spare                                             | 42
   |---------------------------------------------------|
35 | FMAVT index                                       | 43
   |---------------------------------------------------|
36 | Number of read LACBs                              | 44
   |---------------------------------------------------|
37 | Type and disposition                              | 45
   |---------------------------------------------------|
38 | Access mask           | Records per block         | 46
   |---------------------------------------------------|
39 |O|# rd buf  | # wt buf   |er |qw |m |c |d  |s |f   | 47
   |---------------------------------------------------|
40 | Misc. msg file flags                              | 50
   |---------------------------------------------------|
41 | Number of free word in the current free record    | 51
   |---------------------------------------------------|
42 | Number of free records                            | 52
   |---------                             -------------|
43 |                                                   | 53
   |---------------------------------------------------|
44 | Number of nondata records in the file             | 54
   |----                                          -----|
45 |                                                   | 55
   |---------------------------------------------------|
46 | Spare                                             | 56
   |---------------------------------------------------|
47 | #open records     | # read requests               | 57
   |---------------------------------------------------|
48 | last read error   | last write error              | 60
   |---------------------------------------------------|
49 | DST relative address of the next write record     | 61
   |---------------------------------------------------|
50 | Spare                                             | 62
   |---------------------------------------------------|
51 | Spare                                             | 63
   |---------------------------------------------------|
52 | DST rel address of the PACB                       | 64
   |---------------------------------------------------|
53 | DST rel address of the LACB                       | 65
   |---------------------------------------------------|
54 | DST relative address of the stack ACB             | 66
   |---------------------------------------------------|
55 | Stack DST relative address of DB                  | 67
   |---------------------------------------------------|
56 | Target area's DST number                          | 70
   |---------------------------------------------------|
57 | Reserved for calling parameters                   | 71
   |---------------                         ----------|
58 |                                                   | 72
   |---------------                         ----------|
59 |                                                   | 73
   |---------------------------------------------------|
   |---------------------------------------------------|
60 | Reserved for the stack marker from file system    | 74
   |---------------                         ----------|
61 | intrinsics                                        | 75
   |---------------------------------------------------|
   \                                                   \
   |---------------------------------------------------|
64 | User's soft interrupt plabel                      | 100*
   |---------------------------------------------------|
65 | Number of seconds to wait on boundary condition   | 101*
```

```
   |---------------------------------------------------|
66 | O|Ex|Nd|Vr|Bt|Cls  |C  | Carriage control          | 102*
   |---------------------------------------------------|
67 | Reply Port (basic IPC port)                       | 103*
   |---------------------------------------------------|
68 | Writer ID                                         | 104*
   |---------------------------------------------------|
69 | Control block index for nowait writer record buf  | 105*
   |---------------------------------------------------|
70 | DST relative addr of nowait writer record buffer  | 106*
   |---------------------------------------------------|
71 |                                                   | 107*
   |---------------------------------------------------|
72 | No wait I/O resultant error code                  | 110*
   |---------------------------------------------------|
73 | No wait I/O resultant transmission log            | 111
   |---------------------------------------------------|
74 | write wait queue (basic IPC port)                 | 112
   |---------------------------------------------------|
75 | Read wait queue (basic IPC port)                  | 113
   |---------------------------------------------------|
76 | Length of record in bytes                         | 114
   |---------------------------------------------------|
77 | Head record's record type (same values as header)| 115
   |---------------------------------------------------|
   |---------------------------------------------------|
78 | Head record's writer ID                           | 116
   |---------------------------------------------------|
79 | Misc. flags        | Record type                  | 117
   |---------------------------------------------------|
80 | Size of record + count + header words             | 120
   |---------------------------------------------------|
81 | Completor ID       | Waiter ID                    | 121
   |---------------------------------------------------|
82 | Local flags                                       | 122
   |---------------------------------------------------|
83 | Target DST number                                 | 123
   |---------------------------------------------------|
84 | DST relative address of target area               | 124
   |---------------------------------------------------|
85 | Length of target area                             | 125
   |---------------------------------------------------|
86 | Waiter's reply port, 0 if using ACB compltn area  | 126
   |---------------------------------------------------|
87 | Waiting process's PIN                             | 127
   |---------------------------------------------------|
88 | Waiting process's pin                             | 130
   |---------------------------------------------------|
89 | Waiter's soft interrupt plabel                    | 131
   |---------------------------------------------------|
90 | Resultant error code                              | 132
   |---------------------------------------------------|
91 | Resultant transmission log                        | 133
   |---------------------------------------------------|
92 | DST rel address of first buffer                   | 134
   |---------------------------------------------------|
```

```
   | DST rel address of buffer two                     |
   |---------------------------------------------------|
                  .
                  .
                  .
```

* Value is private to a particular accessor.

```
Word   Field   Description
------ ------- ------------------------------------------------

66             Accessor's local flags.
       (0:1)   O  1 -  have not yet issued an FREAD/FWRITE against
                       the file.
       (1:1)   ex 1 -  extended wait mode.
       (2:1)   nd 1 -  do not destroy the next record read.
       (3:1)   vr 1 -  writer has not yet written his first record
                       (ie., he is a virgin).
       (4:1)   bt 0 -  transmission log should be expressed in words.
                  1 -     "        "    "   "     "    " bytes.
       (5:1)   cls  -  Not currently used (reserved for group IPC
                       standard).
       (6:1)   C    -  No wait completion message is in LACB area.
       (8:8) car ctl-  carriage control character to be used for
                       the writer's record (a value of one indicates no
                       carriage control character).
```

```
Word   Field   Description
------ ------- ------------------------------------------------
40             File's global flags.

        (1:4)        - number of read buffers
        (5:4)        - number of write buffers
        (9:1)   er 1 - extended read
        (10:1)  qw 1 - one or more writers has been queued on the

                       wait queue.
        (11:1)  m  1 - wait msg is located in the ACB

        (12:1)  c  1 - completion msg is located in the ACB

        (13:1)  d  1 - the current write buffer has dirty bit set

        (14:1)  s  1 - the start of file is block zero

        (15:1)  f  0 - the ACB buffers have not been filled
```

MMSTAT Definitions

Octal Value	Event Type	Parameter 1	Parameter 2
72/0	Read init	# free rec	
72/1	Read compl	(0:8) error, (8:8) ID	Number of records
72/2	Write init	(0:8) # rec, (8:8) ID	Number of free records
72/3	Write compl	(0:8) error, (8:8) ID	Number of free records
72/4	Control	(0:8) error, (8:8) ID	(0:4) func, (4:12) parm
72/5	EOF	(0:8) error, (8:8) ID	Number of records
72/6	Open	(0:8) error, (8:8) ID	Number of records
72/7	Close	(8:8) #free, (8:8) ID	Number of records
72/10	Initiation	0	(0:8) fix, (8:8) update
73/0	Put record	(0:8) error, (8:8) ID	(0:3) rec type, (3:13) number of records
73/1	Delete rec	(0:8) error, (8:8) ID	(0:3) rec type (3:13) number of records
73/2	Delete blk	Start of file block #	End of file block #

Notes:

1. The aa/bb notation in the "octal value" column denotes type/subtype. Type is the actual MMSTAT event number. Subtype is (0|4) of parameter 0.

2. Several items can possibly exceed their fields, in that case the bits beyond the field are lost. These items are number of records, number of free records, start of file, and end of file.

3. Parameter word zero has a common format for all the MMSTAT events.

Field	Description
(0:4)	Event's subtype.
(4:2)	File's state 0 - empty 1 - partially full 2 - only a fraction of a free record is left 3 - completely full
(6:1)	Nonzero indicates that there is one or more waiting readers.
(7:1)	Nonzero indicates that there is one or more waiting writers.
(11:1)	Nonzero indicates that the write has a carriage control character.
(12:4)	Flags local to the accessor. (12:1) - the accessor has done no FREADs/FWRITEs (13:1) - extended wait (14:1) - nondestructive read (15:1) - writer has not written any records

File System Basic IPC Definitions

The objective of this set of uncallable procedures is to provide a simple ipc mechanism to support the ipc file access procedures. It enables one process to send short, control messages to another process.

General Behavior

FCPORTOPEN Procedure

The heart of this mechanism is the port. A process desiring to receive messages would first open (create) a port. This process is termed the "port manager." When the port is created, a port number is returned to the opener. Since the port number value cannot be known in advance, potential senders need some method of obtaining the port number from the port manager.

Both the ports and the messages are contained in a single disc resident data segment. There can be a total of over thiry-five hundred open ports and outstanding messages Thus neither ports nor message blocks are scarce resources.

FCPORTSEND Procedure

This procedure sends a 0 to 5 word message to a port. Optionally a timeout value may be specified which will limit the duration the message will remain attached to the port. Expiration of the timeout causes the message to be deleted from the target port's queue and placed on the sender's reply port (specified by the sender in the FCPORTSEND procedure call).

{FCPORTRECEIVE}

Reads and deletes the head message from a port. The sender's return port number is also given to the receiver, enabling him to send a reply message.

{FCPORTCLOSE}

Demolishes the port.

{IPC file's use of this mechanism}

All open message files have two ports open for the file (read wait queue and write wait queue), plus one port per accessor (reply port). Their use is described in the following.

Reader and writer wait queues} R When an empty message file is accessed by more than one reader (share), then there must be a way of having the readers' FREADs satisfied in the same order that they were issued. That is, there must be queue of waiting readers. The ipc access procedures accomplish this by dedicating a basic ipc port as a "read wait queue." Whenever a reader's request is stalled because the file is empty, a message is sent to the read wait queue. Subsequent FREADs by other processes will queue up behind the first reader in a FIFO manner. An FWRITE will take the first entry from the wait queue and send a "read may be done" message to the reader's reply port.

In a like manner multiple writers will queue on the write wait queue when the file is full.

{Completion notification for nowait I/O}

The IOWAIT intrinsic waits for a message to be sent to the reply port (s) of the specified user files.

{Timeouts}

When an accessor encounters a boundary condition (ex, a reader accesses an empty file), it may specify that the condition must be satisfied in x seconds (FCONTROL 4). To this end the ipc access procedures merely issue the FCPORTSEND to the wait queue with the user's timeout value specified. The timeout will tear the message from the wait queue and place it on the accessor's reply port.

Port Data Structures

Port Data Segment

```
                        ------------
  System DB extension  |Port DST #| -------
  + %100                ------------       |
                                           |
                                           |
                                           |
                                           |
                      ----------------     |
                     |               |<-----|
  Port data segment  | Global area   |
                     |               |
                     \               \
                     |---------------|
                     |               |
                     | Remainder is  |
                     | composed of   |
                     | "block size"  |
                     | chunks.       |
                     |---------------|
```

The chunks are a combination of free entries, ports, message queue entries, and timer list entries.

Port With Two Outstanding Messages

```
  -------------     ------------      ------------
 |             |----->|          |----->|          |-------
 | Port        |     | MQE 1      |     | MQE 2     |     .
 |             |     |            |     |           |   -----
  -------------     ------------      ------------       ---
                                                          -
```

Port Number

```
  0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---
|Port index |  Port data segment relative addr/8    |
----------------------------------------------------|

  Port index    Index into the port DST number array
```

Port DST Number Array

Located in System DB Extension Area.

```
    ---------------------------------------------------|
64  | Port data segment number                         | 64
    ---------------------------------------------------|
65  | Reserved for a second port segment               | 65
    ---------------------------------------------------|
```

Port Data Segment Global Area

```
     -----------------------------------------------|
 0  |  Data segment number of this port data segment | 0
     -----------------------------------------------|
 1  |  Block size in words                           | 1
     -----------------------------------------------|
 2  |  Total number of blocks                        | 2
     -----------------------------------------------|
 3  |  Maximum number of blocks                      | 3
     -----------------------------------------------|
 4  |  Current number of free blocks                 | 4
     -----------------------------------------------|
 5  |  Number of open ports                          | 5
     -----------------------------------------------|
 6  |  Head of free list                             | 6
     -----------------------------------------------|
 7  |  Tail of free list                             | 7
     -----------------------------------------------|
10  |  Head of impeded process list                  | 8
     -----------------------------------------------|
11  |  Tail of impeded process list                  | 9
     -----------------------------------------------|
12  |  Head of timeout thread (TQE address)          | 10
     -----------------------------------------------|
13  |  TRLX of timeout                               | 11
     -----------------------------------------------|
14  |  Value returned by TIMER intrinsic when        | 12
     -------------------     -----------------------
15  |  Timeout was initiated.                        | 13
     -----------------------------------------------|
16  |  Head of port list (in units of port numbers). | 14
     -----------------------------------------------|
17  |  Not used.                                     | 15
     -----------------------------------------------|
```

Port

```
    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
   ---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---
 0 |  Head MQE address                              | 0
   ------------------------------------------------
 1 |  Tail MQE address                              | 1
   ------------------------------------------------
 2 |E |  W  |  Next port number in port list thread| 2
   ------------------------------------------------
 3 |I |Subtype|Port Pin number                      | 3
   ------------------------------------------------
 4 |  Soft interrupt parameter one                  | 4
   ------------------------------------------------
 5 |  Number of MQEs in the port's queue            | 5
   ------------------------------------------------
 6 |  Number of sends to this port                  | 6
   ------------------------------------------------
 7 |  Soft interrupt plabel                         | 7
   ------------------------------------------------
 8 | PIN of port's owner                            | 10
   ------------------------------------------------
   |0 |1 |2 |3 |4 |5 |6 |7 |8 |9 |10|11|12|13|14|15|
```

```
E        Enable wake up bit
         0 - Do not awaken the process
         1 - Awaken the process

W type   Action to be taken on an enabled port when a message is
         received.

         0 - Awaken the process on a message wait bit.

         1 - Generate user software interrupt

         2 - Generate system software interrupt

I         Interrupt mode.

Subtype  Soft interrupt subtype
```

Message Queue Entry (MQE)

```
    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
   ---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---
 0 |  Next MQE entry; if last, (port addr) LOR 7   | 0
   ------------------------------------------------
 1 |  Port number of return port                   | 1
   ------------------------------------------------
 2 |Time List Entry (TLE),0=no timeout,-1=timed out| 2
   ------------------------------------------------
 3 |  Parameter zero                               | 3
   ------------------------------------------------
 4 |  Parameter one                                | 4
   ------------------------------------------------
 5 |  Parameter two                                | 5
   ------------------------------------------------
 6 |  Parameter three                              | 6
   ------------------------------------------------
 7 |  Parameter four                               | 7
   ------------------------------------------------
   |0 |1 |2 |3 |4 |5 |6 |7 |8 |9 |10|11|12|13|14|15|
```

Timer entry definitions - 0 - no timeout
1 - timeout expired
2 - TLE address for a pending timeout

File System Message Files

Wait Message

parm#
0 - WRITER ID
1 - LOCAL FLAGS (differ with each accessor)
(0:1) - accessor just opened file
(1:1) - will wait on boundary condition if no symbiotic process
(3:1) - writer has not written a record
(4:1) - transmission log in bytes
(8:1) - carriage control code
2 - DST# of data buffer
3 - Address of data buffer (DST relative)
4 - Length of data buffer in bytes

Completion Message

0 - Resultant error code
1 - Resultant transmission log in bytes

Timer List Entry (TLE)

```
    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
   ---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---
 0 |  Next TLE (sorted in incr time val), 0 if last| 0
   ------------------------------------------------
 1 |  Preceding TLE entry (0 if first entry)       | 1
   ------------------------------------------------
 2 |  Number of milliseconds the timeout value     | 2
   ----------------               -----------------
 3 |  of this TLE is beyond the previous TLE.      | 3
   ------------------------------------------------
 4 |  Address of the affected MQE                  | 4
   ------------------------------------------------
 5 |  Address of the MQE's port                    | 5
   ------------------------------------------------
 6 |  Value of TIMER when this timeout expires     | 5
   -------------                     ---------------
 7 |  (Milliseconds)                               | 7
   ------------------------------------------------
   |0 |1 |2 |3 |4 |5 |6 |7 |8 |9 |10|11|12|13|14|15|
```

MMSTAT Definitions

Octal Value	Event Type	Parameter 0	Parameter 1	Parameter 2
62	Open	Port number	Port DST num	Flags parameter
63	Receive completion	Port number	MQE address 15:1 Waitspc	Return port
64	Send	Port number	MQE address 15:1 Q type	Return port
65	Change status	Port number	0 = enable 1 = disable	Head MQE address
66	Abort	Port number	Parameter zero	Return port
67	Close	Port number	Port DST	# open ports left
70	Expand	Port DST num	# expand blks	Total # blocks
71	Timeout expired	Port num	MQE address	Return port

CHAPTER 19 MPE MEMORY RESIDENT MESSAGE FACILITY

Overview of Facility

The memory resident message facility of MPE V addresses the need for an efficient, simple, and uniform method for system code to send short status-type messages to processes.

Each process is created with a "port" in the message harbor table (DST %71) which supports a set of message subqueues which are private to that process. There is a maximum of four subqueues per port in the initial implementation. This limit can be easily extended when new subqueues are required.

Any system code, even code running on the ICS, can send a message to any subqueue of any process. The destination process' PIN must be known, any a priori conventions on subqueue number and message formats must be established. The caller of SENDMSG may optionally specify that the destination process be awakended from a message wait.

Message can be any length up to the configured maximum. Message length is specified in the call to SENDMSG and RECEIVEMSG. In the initial implementation, messages are limited to 4 words in length. This maximum can easily be increased if the need arises.

By calling PORTSTATUS, a process may at any time determine whether a specified subqueue is non-empty or obtain the subqueue number of the most urgent non-empty subqueue (lowest numbered one).

By calling RECEIVEMSG, a process may receive the message at the head of the specified subqueue. This receive is optionally non-destructive.

A process can wait on a message wait, or on a combination of message wait and other wait types.

Message Intrinsics

SENDMSG

```
        Procedure SENDMSG(Destpin, Subqueue, MsgLength, Flags);
        Value           Destpin, Subqueue, MsgLength, Flags;
        Integer         Destpin, Subqueue, MsgLength;
        Logical                                       Flags;
        Option Privileged, Uncallable;
```

Destpin, Subqueue, and MsgLength have to be within range or a System Failure 622 will occur.

The caller of SENDMSG stacks the message contents before calling the procedure. SENDMSG expects the first msg word to be at Q-7-MsgLength, and the last msg word at Q-8. The message contents at Q-8 to Q-7-MsgLength are deleted from the top of stack by the exit from SENDMSG to the caller.

Flags.(1:1) = 1 ==> Wake-up destination process from a message wait.

Return CC = CCG if process was already awake else CC = CCE.

PORTSTATUS

```
        Logical Procedure PORTSTATUS(Subqueue);
        Value                       Subqueue;
        Integer                     Subqueue;
        Option Privileged, Uncallable;
```

When supplied a valid subqueue number, PORTSTATUS returns a true value if the subqueue is non-empty and a false value if the subqueue is empty.

When passed a -1 a subqueue parameter, PORTSTATUS returns the subqueue number of the process' most urgent non-empty subqueue (the smaller the number, the more urgent the subqueue).

If all subqueues are empty, PORTSTATUS returns CC - CCE. If at least one subqueue is non-empty, PORTSTATUS returns CC = CCG.

RECEIVEMSG

```
        Procedure RECEIVEMSG(Subqueue, MsgLength, Flags);
        Value             Subqueue, MsgLength, Flags;
        Integer           Subqueue, MsgLength;
        Logical                                Flags;
        Option Privileged, Uncallable;
```

Subqueue and MsgLength has better be within range or a System Failure 622 will occur.

The caller of RECEIVEMSG does an ASSEMBLE(ADDS MsgLength) to make space for the message contents. RECEIVEMSG stores the message contents into Q-8, Q-9,...,Q-7-MsgLength. Q-7-MsgLength contains the first word of the message.

Flags.(0:1) ==> do not release message from head of subqueue (non destructive read).

Return CC = CCG if all subqueues were empty, else CC = CCE.

Supporting Data Structures

Message Harbor Table [DST #57 (%71)]

```
    +------------------------------+
  0 | DST Index Number  (%71)      |
    +------------------------------+
  1 | Data Segment Size            |
    +------------------------------+
  2 | Reserved                     |
    +------------------------------+
  3 | Maximum number of PINS + 1   |
    +------------------------------+
  4 | Maximum Msg Size (6)         |
    +------------------------------+
  5 | Reserved                     |
    +------------------------------+
  6 | Message Pool Head Pointer    |
    +------------------------------+
  7 | Message Pool Tail Pointer    |
    +------------------------------+
  8 | Available Msg Frames Count   |
    +------------------------------+
  9 | Head of impeded queue        |
    +------------------------------+
 10 | Tail of impeded queue        |
    +------------------------------+
 11 | Reserved                     |
    |                              |
    +------------------------------+
 13 | Ports     (16 words each)    |
    | (8 for header + 2 link words |
    |  for each of 4 subqueues)    |
    +------------------------------+
    | Messages  (6 words each)     |
    | (2 for header + 4 for data)  |
    +------------------------------+
```

CHAPTER 20 MMSTATS EVENTS

MMSTATS Catalog Index

EVENT NAME	EVENT NO.			EVENT NAME	EVENT NO.	
	DEC.	%			DEC.	%
-----------	---------			----------	---------	
ALCSTBLK	20	024 (-)	*	FREAD	62	076 (-)
ALLOCMEM	12	014	*	FREADDIR	64	100 (-)
BINREAD	233	351 (-)	*	FREADLABEL	76	114 (-)
BREAK	237	355 (-)	*	FREADSEEK	68	104 (-)
C_ABSENT	139	213	*			
CABORTIO	142	216	*	FRENAME	80	120 (-)
CACHEMOV	14	016	*			
CCLOSE	146	222	*	FSETMODE	72	110 (-)
CCLOSETRACEFILE	154	232	*	FSPACE	69	105 (-)
CCONTROL	152	230	*	FUNLOCK	79	117 (-)
CDT_ATT	86	126	*			
CGARBAGE	7	007	*	FUPDATE	66	102 (-)
CONFIG-INFO	221	335 (-)	*	FWRITE	63	077 (-)
CONFIG-INFO	222	336 (-)	*	FWRITEDIR	65	101 (-)
CONFIG-INFO	223	337 (-)	*	FWRITELABEL	77	115 (-)
COPEN	140	214	*	GIPINTERRUPT	192	300
			*	GET_CDT	15	017
COPENTRACEFILE	153	231	*	IOBUFTRAP	125	175
CPOLLIST	155	233	*	I/O COMPLETION	111	157 (-)
			*	INITIATE	84	124
CREAD	147	223	*	IOWAIT	67	103 (-)
			*	LINK_REG	89	131
CREAD1	147	240	*	MAKEOC	1	001
			*	MAP_DOM	87	127
CSDRIVER	150	226	*	MONINIT	228	344 (-)
CSIOWAIT	144	220	*	MONOFF	229	345 (-)
CWRITE	149	225	*	PROCESS COMPLETE	211	323 (-)
DC1DC2ACK	231	347 (-)	*	QONSEG	0	000
			*	QUE_LDR	16	020
DEALLOCM	13	015	*	QUIESCE	40	050
DEALCSTBLK	21	025 (-)	*	RELRESOURCES	23	027 (-)
			*	REQCACHE	90	132
DISKBUGCATCHER	200	310	*	SEGIOINIT	5	005
			*	SIODM-ENTRY	194	302
DISKBUGCATCHER	201	311	*	SIODM	195	303
DISKERROR	100	144 (-)	*	SIODONE	6	006
			*	SOFT'DEATH	120	170
DISKERROR	101	145 (-)	*	SPECCHAR	236	354 (-)
DISKINTRPT	191	277	*	SPECIALRQ	2	002
DQUE_LDR	17	021	*			
			*	SPECREAD	238	356 (-)
			*	START I/O	193	301
			*	STRATEGY	83	123
DISK TRAFFIC	98	142 (-)	*	SWAPIN	8	010
FCHECK	74	112 (-)	*	SYSPINS	224	340 (-)

FCLOSE	81	121 (-)	*	SYSPINS	225	341 (-)
FCONTROL	71	107 (-)	*	SYSPINS	226	342 (-)
FETCHSEG	4	004	*	SYSPINS	227	343 (-)
FGETINFO	75	113 (-)	*	TERMLOGOFF	235	353 (-)
FIND_DE	18	022	*			
FLOCK	78	116 (-)	*	TERMLOGON	234	352 (-)
FOPEN/(DA)	60	074 (-)	*	TERMREAD	230	346 (-)
FOPEN/(DA)	61	075 (-)	*	TERMWRITE	232	350 (-)
FPOINT	70	106 (-)	*	UN_MAP_RG	88	130

MMSTAT CATALOG INDEX

EVENT GROUP	DESCRIPTION OF GROUP	PAGE NO.
-----	-----------	------
0	MEMORY MANAGER	20-1
1	MEMORY MANAGER/CACHING	20-9
2	MEMORY MANAGER	20-10
4	SCHEDULING	20-13
6	FILESYS	20-16
7	FILESYS	20-25
8	FILESYS/CACHING	20-30
9	DISC I/O TRANSFER/CACHING	20-31
10	DISC ERRORS	20-32
11	SIO	20-33
12	DISC SPACE	20-34
13	DISC CACHING	20-51
14	CS/3000	20-36
15	CS/3000	20-40
16	CS/3000	20-43
19	DISC CONTROLLER INTRPT	20-44
20	PRIVATE VOLUMES	20-47
21	PROCESS CREATION AND TERMINATION	20-48
22	MONITOR CONFIG INFORMATION	20-49
23	TERMINAL I/O	20-53

MMSTAT Event Group 0 (Memory Management Events)

Event 0

EVENT NAME: QONSEG
DESCRIPTION: ABSENCE TRAP ON CODE/DATA SEGMENT

CALLING MODULE: KERNELC
CALLING PROCEDURE(S): QUEUEONSEGMENT

PARAMETER DESCRIPTION

P1,P2 = Segment Identifier

```
P1.(0:4)  = Segment type field
              0   => Data Segment
              1   => SL Segment
              2   => Program Segment
              3   => Cache Domain

P1.(4:12) = Program index into CSTBLK (type 2 only)

P2        = Segment Number
```

P3 = SLL Pointer (SLL table relative)

P4 = STATUS (in stack marker) of calling (trapping) segment

P5,P6 - Unused.

Event 1

```
EVENT NAME:  MAKEOC
DESCRIPTION: MAKE SEGMENT AN OVERLAY CANDIDATE - RELEASE SEGMENT
             TO THE POOL OF AVAILABLE SPACE

CALLING MODULE:  KERNELC
CALLING PROCEDURE:  MAKEOC

     PARAMETER DESCRIPTION
     ---------------------

     P1,P2 = Segment Identifier

             P1.(0:4)  = Segment type field
                           0   => Data Segment
                           1   => SL Segment
                           2   => Program Segment
                           3   => Cache Domain

             P1.(4:12) = Program index into CSTBLK (type 2 only)

             P2        = Segment Number


     P3   =   Bank of region
     P4   =   Address of region

     P5,P6 - Unused.
```

Event 2

```
EVENT NAME:  SPECIALRQ
DESCRIPTION:  REQUEST OF SEGMENT EXPANSION/CONTRACTION, UNLOCK,
              UNFREEZE, IOUNFREEZE, LOCK, IOFREEZE, FREEZE

CALLING MODULE:  KERNELC, KERNELD, ININ
CALLING PROCEDURES:  UNLOCKSEG', IOFREEZE', FETCHSEGMENT-(KERNELC)
                     DLSIZE, ZSIZE, GETPXSEG, ALTDSEGSIZE,
                     ALTPXFILESIZE                      -(KERNELD)
                     STACKOVERFLOW                      -(ININ)

     PARAMETER DESCRIPTION
     ---------------------

     P1,P2 = Segment Identifier

             P1.(0:4)  = Segment type field
                           0   => Data Segment
                           1   => SL Segment
                           2   => Program Segment
                           3   => Cache Domain

             P1.(4:12) = Program index into CSTBLK (type 2 only)

             P2        = Segment Number

     P3 = .(0:1)   =1 => Request is through FETCHSEGMENT
                    (types 0,1,2)
          .(12:4)   Type of request
                   =  0=> IOFREEZE
                   =  1=> FREEZE
                   =  2=> LOCK
                   =  3=> IOUNFREEZE
                   =  4=> UNFREEZE
                   =  5=> UNLOCK
                   =  6=> DLSIZE EXPANSION
                   =  7=> DLSIZE CONTRACTION
                   =  8=> PXFIXED EXPANSION
                   =  9=> PXFILE EXPANSION
                   = 10=> PXFILE CONTRACTION
                   = 11=> XDS EXPANSION
                   = 12=> XDS CONTRACTION
                   = 13=> ZSIZE EXPANSION
                   = 14=> ZSIZE CONTRACTION
                   = 15=> STACKOVERFLOW

     P4 = For types (P3.(12:4))
            = 0,2,3,5 => P4.(8:8) = LOCK OR IOFREEZE COUNT
            = 1,4 => P4.(0:8) = FREEZE COUNT
            = 6-15 => REQUESTED SIZE OF AREA IN WORDS

     P5,P6 - Unused.
```

Event 4

```
EVENT NAME:  FETCHSEG
DESCRIPTION:  SEGMENT REQUEST (FOR I/O SYSTEM OR PROCESS)

CALLING MODULE:  KERNELC
CALLING PROCEDURE:  FETCHSEGMENT

     PARAMETER DESCRIPTION
     ---------------------

     P1,P2 = Segment Identifier

             P1.(0:4)  = Segment type field
                           0   => Data Segment
                           1   => SL Segment
                           2   => Program Segment
                           3   => Cache Domain
             P1.(4:12) = Program index into CSTBLK (type 2 only)

             P2        = Segment Number

     P3 = Requester ID
          .(0:1) = 1   => I/O System request
                          .(1:15) = Ldev #
          .(0:1) = 0   => Process request
                          .(1:15) = Pin # of requesting process

          .(1:1) = 1   => IOFREEZE REQUEST
          .(2:1) = 1   => BLOCKED LOCK REQUEST
          .(3:1) = 1   => LOCK REQUEST
          .(4:1) = 1   => FREEZE REQUEST

     P4=  .(13:3)= 0   => Segment already present
                 = 1   => Segment is Recover Overlay Candidate
                 = 2   => Segment already on its way in for someone
                          (Segment In Motion In)
                 = 3   => Segment not present -- must fetch
                          (Full fetch)

     P5,P6 - Unused.
```

Event 5

```
EVENT NAME:  SEGIO
DESCRIPTION:  MEMORY MANAGEMENT READ/WRITE OF SEGMENT FROM/TO
              DISC QUEUED

CALLING MODULE:  KERNELC
CALLING PROCEDURES:  PROCESSINITMSG, STARTSEGWRITE

     PARAMETER DESCRIPTION
     ---------------------
     P1,P2 = Segment Identifier

             P1.(0:4)  = Segment type field
                           0   => Data Segment
                           1   => SL Segment
                           2   => Program Segment
                           3   => Cache Domain

             P1.(4:12) = Program index into CSTBLK (type 2 only)

             P2        = Segment Number

     P3 = Disc Request Index - (DRQ Table relative)

     P4 = .(0:1) = 1  => WRITE START
                 = 0  => READ START
          .(1:15)= Ldev #

     P5,P6 - Unused.
```

Event 6

```
EVENT NAME:  SIODONE
DESCRIPTION:  MEMORY MANAGEMENT SEGMENT READ/WRITE FROM/TO DISC
              COMPLETE

CALLING MODULE:  KERNELC
CALLING PROCEDURES:  SEGREADCOMPLETOR, SEGWRITECOMPLETOR

     PARAMETER DESCRIPTION
     ---------------------

     P1,P2 = Segment Identifier

             P1.(0:4)  = Segment type field
                         0   => Data Segment
                         1   => SL Segment
                         2   => Program Segment
                         3   => Cache Domain

             P1.(4:12) = Program index into CSTBLK (type 2 only)

             P2        = Segment Number

     P3 = Disc Request Index  (DRQ Table relative)
     P4 = .(0:1) = 1  => Write complete
                 = 0  => Read complete

     P5,P6 - Unused.
```

Event 7 (%7)

```
EVENT NAME:  CGARBAGE
EVENT DESCRIPTION:  GARBAGE COLLECTION HAS JUST TAKEN PLACE

CALLING MODULE:  KERNELC
CALLING PROCEDURE:  COLLECTGARBAGE

     PARAMETER DESCRIPTION
     ---------------------
     P1 = BANK OF SOURCE JUST MOVED FROM
     P2 = ADDR OF SOURCE JUST MOVED FROM
     P3 = MOVEPAGECNT, NUMBER OF PAGES JUST MOVED FROM
     P4,P5,P6 - Unused.
```

Event 8 (%10)

```
EVENT NAME:  SWAPIN
DESCRIPTION:  SWAP IN A PROCESS

CALLING MODULE:  KERNELC
CALLING PROCEDURE:  SWAPIN

     PARAMETER DESCRIPTION
     ---------------------

     P1 = PIN OF PROCESS BEING SWAPPED IN
     P2 = .(0:1) =  0 => BEING SWAP
                 =  1 => END SWAP
          .(1:1) =  0 => NORMAL (PARTIAL SWAP OK)
                 =  1 => SWAP REQUIRED
          .(12:4)=  0 => PROCESS SWAPIN COMPLETE
                    2 => NO ROOM, HARD REQ MAY SUCCEED
                    3 => NO ROOM, HARD REQ FAILED
                    4 => SWAPIN STOPPED - MORE URGENT ACTIVITY
                    8 => NO LOCK SPACE
     P3 = HARDREQUEST = TRUE => HARD REQUEST ON SWAPIN
                        FALSE=> NORMAL

     P4,P5,P6 - Unused.
```

MMSTAT Event Group 1 (Memory Manager)

Event 12 (%14)

```
EVENT NAME:  ALLOCMEM
DESCRIPTION:  FOUND A HOLE FOR A SEGMENT REPLACEMENT REQUEST

CALLING MODULE:  KERNELC
CALLING PROCEDURE:  RESERVEREGION

     PARAMETER DESCRIPTION
     ---------------------
     P1 = REQUESTED SIZE IN PAGES
     P2 = BANK OF SELECTED REGION
     P3 = ADDRESS OF SELECTED REGION
     P4,P5,P6 - Unused.
```

Event 13 (%15)

```
EVENT NAME:  DEALLOCM
DESCRIPTION:  RELEASE REGION OF MEMORY TO AVAILABLE STATUS

CALLING MODULE:  KERNELC
CALLING PROCEDURE:  RELEASEREGION

     PARAMETER DESCRIPTION
     ---------------------

     P1 = SIZE RELEASED IN PAGES
     P2 = BANK OF RELEASED REGION BASE
     P3 = ADDRESS OF RELEASED REGION BASE
     P4,P5,P6 - Unused.
```

Event 14 (%16)

```
Event Name: CACHEMOV
Description: A cache move (i.e. logical disc request) has
             just completed.
Calling Module: CACHESEG
Calling Procedure: ProcessCDTLogReqQue

Parameter Description
---------------------

P1,P2  = Segment identifier of target DST (LDR'BUFDST)
         P2.(0:1) = 1 then this is a stack.
P3     = Mapped Domain CDT entry number
P4     = Transfer count
P5,P6  = Unused
```

Event 15 (%17)

```
Event Name: GET_CDT
Description: Called when an entry in the CDT table is
             obtained or released.
Calling Module: CACHESEG
Calling Procedures: Get'CDT'Entry, CDT'Free'Entry,
                    CDT'Get'MD'Entry, CDT'Rel'MD'Entry

Parameter Description
---------------------

P1      = CDT entry number
P2      = Type of call
          0 = Free entry
          1 = Get entry
          2 = Get Mapped Domain entry
          3 = Release Mapped Domain entry
P3      = If P2=3 then Ldev Entry number
P4,P5,P6  Not used.
```

Event 16 (%20)

Event Name: QUE_LDR
Description: Called when an LDR is queued onto the CDT
Calling Module: CACHESEG
Calling Procedure: CDT'Queue'LDR

```
Parameter Description
---------------------

P1     = Mapped Domain CDT entry number
P2     = LDR entry index to be queued
P3     = Queue type
         %12 - CDT impeded queue
         %13 - CDT active queue
P4,P5,P6  Not used.
```

Event 17 (%21)

Event Name: DQUE_LDR
Description: Called when an LDR is removed from the CDT queue.
Calling Module: CACHESEG
Calling Procedure: CDT'Dequeue'LDR

```
Parameter Description
---------------------

P1     = Mapped Domain CDT entry number
P2     = LDR entry index being removed from the queue
P3     = Queue type
         %12 - CDT impeded queue
         %13 - CDT active queue
P4,P5,P6 Not used.
```

Event 18 (%22)

Event Name: FIND_DE
Description: Called when need to find an assigned CDT
Device entry.
Calling Module: CACHESEG
Calling Procedure: CDT'Find'DE

```
Parameter Description
---------------------

P1     = Ldev number of the CDT Device entry to be found.
P2     = CDT Device entry
P3,P4,P5,P6 Not used.
```

MMSTAT Event Group 2

Event -20 (-%24)

EVENT NAME: ALCSTBLK
DESCRIPTION: REQUEST TO RESERVE A BLOCK OF ENTRIES IN THE CSTX

CALLING MODULE: KERNELD
CALLING PROCEDURE: ALCSTBLOCK

```
PARAMETER DESCRIPTION
---------------------

P1=EIX     CST BLOCK INDEX ASSIGNED
P2=CSTX    DST RELATIVE INDEX OF WORD 0
           OF THE FIRST RESERVED CSTX ENTRY
P3=N       NUMBER OF CSTX ENTRIES RESERVED
P4,P5,P6 - Unused.
```

Event -21 (%25)

EVENT NAME: DEALCSTBLK
DESCRIPTION: INDICATES THAT A CST EXTENSION BLOCK HAS BEEN
DEALLOCATED

CALLING MODULE: KERNELD
CALLING PROCEDURE: DEALCSTBLOCK

```
PARAMETERS      PARAMETER DESCRIPTION

    P1=EIX     CST BLOCK INDEX ASSIGNED
               TO THE BLOCK OF CST ENTRIES
    P2=CSTX    DST RELATIVE INDEX OF WORD 0
               OF THE FIRST CST ENTRY TO BE
               RELEASED
    P3=MCNT    =(#ALLOCATED CSTX ENTRIES-
                 #ENTRIES BEING RELEASED)*4
    P4,P5,P6 - Unused.
```

Event -23 (-%27)

EVENT NAME:RELRESOURCES
DESCRIPTION: RESOURCES (VDS,MAIN MEMORY, ST ENTRY) RESERVED FOR THE
FOR THE SEGMENT HAVE BEEN RELEASED

CALLING MODULE: KERNELD

CALLING PROCEDURE: RELDATASEG

```
PARAMETERS      PARAMETER DESCRIPTION

    P1=NEW DB DST NUMBER
    P2=DELTA P AT EXCHANGEDB CALL

    P3=STATUS AT EXCHANGEDB CALL
    P4,P5,P6 - Unused.
```

MMMSTAT Event Group 3

(NOT CURRENTLY ASSIGNED)

MMSTAT Event Group 4 (Scheduling)

Event 40 (%50)

EVENT NAME: QUIESCE
DESCRIPTION: PROCESS SWITCH - STATE OF PROCESS SAVED

CALLING MODULE: KERNELC
CALLING PROCEDURE: DSP

```
     PARAMETER DESCRIPTION
     ---------------------

     P1 = PCB00(CPCB)
          .(0:1) =  1  => SAR  - SCHEDULING ATTENTION REQUIRED
          .(2:1) =  1  => CRIT - PROCESS IS CRITICAL
          .(3:1) =  1  => HSIR - PROCESS HAS SIR
          .(4:1) =  1  => PIOVR - PENDING PI, PROCESS CRITICAL
          .(5:1) =  1  => HSPRI - HOLD SIR PRIORITY
          .(6:1) =  1  => IPEXP - INCORE PROTECT EXPIRED
          .(7:1) =  1  => PC    - PREEMPT CAPABILITY
          .(8:1) =  1  => MP    - MUST PREEMPT
          .(9:1) =  1  => LW    - LONG WAIT
          .(10:1)=  1  => SW    - SHORT WAIT
          .(11:1)=  1  => TRW   - TERMINAL READ WAIT
          .(12:1) =1  =>  USEQD - USED A QUANTUM SINCE TRANSACTION
                                   BEGAN
          .(13:1)=  1  => HIPRI - HOLD IMPEDED PRIORITY
          .(14:1)=  1  => ALLOW SOFT INTERRUPTS EVEN THOUGH IN
                          SYSTEM CODE
          .(15:1)=  1  => RITBK - PROCESS IN RIT BREAK

P2 = PCB04(CPCB)
          .(0:1) =  1  => M     - MOURNING WAIT
          .(1:1) =  1  => RG    - GLOBAL RIN WAIT
          .(2:1) =  1  => RL    - LOCAL RIN WAIT
          .(3:1) =  1  => MA    - MAIL WAIT
          .(4:1) =  1  => BIO   - BLOCKED IO WAIT
          .(5:1) =  1  => IO    - IO WAIT
          .(6:1) =  1  => UCP   - UCOP WAIT, RIT WAIT
          .(7:1) =  1  => JNK   - JUNK WAIT
          .(8:1) =  1  => TIM   - TIMER WAIT
          .(9:1) =  1  => INT   - INTERRUPT WAIT
          .(10:1)=  1  => SON   - SON WAIT
          .(11:1)=  1  => FA    - FATHER WAIT
          .(12:1)=  1  => IMP   - PROCESS WAITING TO UNIMPEDED
          .(13:1)=  1  => SIR   - PROCESS WAITING FOR SIR
          .(14:1)=  1  => TIM   - PROCESS WAITING FOR TIME OUT
          .(15:1)=  1  => MEM   - PROCESS WAITING FOR MEMORY
```

```
P3 = PCB13(CPCB)
        .(0:1) =  1  => DISPQ - PROCESS ON DISPATCHING QUEUE

        .(1:1) =  1  => L SCHEDULING CLASS
        .(2:1) =  1  => C SCHEDULING CLASS
        .(3:1) =  1  => D SCHEDULING CLASS
        .(4:1) =  1  => E SCHEDULING CLASS
        .(5:1) =  1  => INTER- PROCESS IS INTERACTIVE
        .(6:1) =  1  => CORER- PROCESS IS CORE-RESIDENT
        .(8:8) =  PROCESS' SCHEDULING PRIORITY
```

P4,P5,P6 - Unused.

MMSTAT Event Group 5

(SEE CHAPTER 18 FOR THESE EVENTS)

MMSTAT Event Group 6 (FILESYS)

THESE EVENTS ARE FOR DEVELOPMENT USE ONLY AND ARE NOT NORMALLY ENABLED

Event -60(%74)

EVENT NAME: FOPEN
DESCRIPTION: OLD FILE OPEN

CALLING MODULE: FILEACC

CALLING PROCEDURE: FOPENDA

```
          PARAMETERS      PARAMETER DESCRIPTION

             P1= FILE #      (0:2)=2 -> NON-SPOOLER ACCESS
                             (0:2).NE.2 ->

             P2= AOPTIONS   SEE INTRINSICS MANUAL
             P3= FILE LABEL FOPTIONS   SEE INTRINSICS MANUAL
             P4= RECORD SIZE
             P5= FILE LABEL BLOCK SIZE
             P6= # OF BUFFERS
```

Event -61(%75)

EVENT NAME: FOPEN'
DESCRIPTION: OLD FILE OPEN (CONTINUATION OF EVENT -60)

CALLING MODULE: FILEACC

CALLING PROCEDURE: FOPENDA

```
          PARAMETERS      PARAMETER DESCRIPTION

             P1= FILE LABEL FILE LIMIT      MSW

             P2= FILE LABEL FILE LIMIT      LSW

             P3= FILE LABEL # OF EXTENTS

             P4-P6 unused
```

Event -60(%74)

EVENT NAME: FOPEN
DESCRIPTION: NEW DISC FILE OPEN

CALLING MODULE: FILEACC

CALLING PROCEDURE: FOPEN

```
          PARAMETERS      PARAMETER DESCRIPTION

             P1= FILE #     (0:2)=2 -> NON-SPOOLER ACCESS
                            (0:2).NE.2 ->
             P2= AOPTIONS   SEE INTRINSICS MANUAL

             P3= FOPTIONS   SEE INTRINSICS MANUAL

             P4= RECORD SIZE

             P5= BLOCK SIZE

             P6= # OF BUFFERS
```

Event -61(%75)

EVENT NAME: FOPEN'
DESCRIPTION: NEW DISC FILE OPEN (CONTINUATION OF EVENT -60)

CALLING MODULE: FILEACC

CALLING PROCEDURE: FOPEN

```
          PARAMETERS      PARAMETER DESCRIPTION

             P1= FCB FILE LIMIT

             P2= FCB MAX # EXTENTS

             P3= (0:8)= INITIAL ALLOCATION EXTENTS

             P4-P6 unused
```

Event -62(%76)

EVENT NAME: FREAD
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FREAD

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= ACBTLOG	TRANSFER COUNT
P3= FLAGS	(15:1) Buffer hit flag

Event -63(%77)

EVENT NAME: FWRITE
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FWRITE

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= TCOUNT	SEE INTRINSIC MANUAL
P3= FLAGS	(15:1) Buffer hit flag

Event -64(%100)

EVENT NAME: FREADDIR
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FREADDIR

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= ACBTLOG	TRANSFER COUNT
P3= FLAGS	(15:1) Buffer hit flag
P4= REC #	MSW
P5= REC #	LSW
P6= NOT USED	

Event -65(%101)

EVENT NAME: FWRITEDIR
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING MODULE: FWRITEDIR

PARAMETERS	PARAMETER DESCRIPTION
P1= FILENUM	
P2= TCOUNT	See Intrinsic manual
P3= FLAGS	(15:1) Buffer hit flag
P4= REC #	MSW
P5= REC #	LSW
P6= NOT USED	

Event -66(%102)

EVENT NAME: FUPDATE
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FUPDATE

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= TCOUNT	See Intrinsic manual
P3= FLAGS	(15:1) Buffer hit flag

P4-P6 not used

Event -67(%103)

EVENT NAME: IOWAIT
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: IOWAIT

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= ACBTLOG	TRANSFER COUNT
P3= FLAGS	(15:1) buffer hit flag

Event -68(%104)

EVENT NAME: FREADSEEK
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FREADSEEK

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= FLAGS	(15:1) buffer hit flag
P3= REC #	MSW
P4= REC #	LSW
P5-P6 not used	

Event -69 (%105)

EVENT NAME: FSPACE
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FSPACE

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= DISPLACEMENT	SEE INTRINSIC MANUAL
P3-P6	not used

MMSTAT Event Group 7 (FILESYS)

THESE EVENTS ARE FOR DEVELOPMENT USE ONLY AND ARE NOT NORMALLY ENABLED

Event -70 (%106)

EVENT NAME: FPOINT
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FPOINT

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= REC #	MSW
P3= LSW	LSW
P4-P6	not used

Event -71 (%107)

EVENT NAME: FCONTROL
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FCONTROL

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= CODE	See Intrinsics manual
P3-P6	not used

Event -72 (%110)

EVENT NAME: FSETMODE
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FSETMODE

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= MODEFLAGS	SEE INTRINSIC MANUAL
P3-P6	not used

Event -74 (%112)

EVENT NAME: FCHECK
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FCHECK

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= ERRORCODE	SEE INTRINSIC MANUAL
P3-P6	not used

Event -75 (%113)

EVENT NAME: FGETINFO
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FGETINFO

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= FOPTIONS	SEE INTRINSIC MANUAL
P3= AOPTIONS	SEE INTRINSIC MANUAL
P4-P6	not used

Event -76 (%114)

EVENT NAME: FREADLABEL
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE:

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= TCOUNT	SEE INTRINSIC MANUAL
P3-P6	unused

Event -77 (%115)

EVENT NAME: FWRITELABEL
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FWRITELABEL

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= TCOUNT	SEE INTRINSIC MANUAL
P3-P6	unused

Event -78 (%116)

EVENT NAME: FLOCK
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROLEDURE: FLOCK

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= LOCKCOND	See Intrinsics manual
P3= COND CODE	" " " "

Event -79 (%117)

EVENT NAME: FUNLOCK
DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FUNLOCK

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2-P6	unused

MMSTAT Event Group 8

Event -80 (%120)

EVENT NAME: FRENAME
DESCRIPTION:

CALLING MODULE: FILEACC

CALLING PROCEDURE: FRENAME

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2-P6	unused

Event -81 (%121)

EVENT NAME: FCLOSE
DESCRIPTION:

CALLING MODULE: FILEACC

CALLING PROCEDURE: FCLOSE

PARAMETERS	PARAMETER DESCRIPTION
P1= FILE #	
P2= DISP	See Intrinsic manual
P3= SECCODE	
P4-P6	unused

Event 83 (%123)

Event Name: STRATEGY
Description: Called to determine the type of strategy used
based on who the caller of CDT'ATTACHIO is.
Calling Module: CACHESEG
Calling Procedure: CDT'STRATEGY

Parameter Description

P1 = CDT Mapped Domain entry
P2 = LDR entry index
P3 = Strategy
0 - Unknown caller
1 - Unknown from File System
2 - Spooler
3 - Directory
4-7 - Unknown
8 - Genmessage
9 - File System, Quiesce I/O
10 - File System, sequential, NOBUF
11 - File System, direct, NOBUF
12 - File System, sequential, BUF
13 - File System, direct, BUF
14 - File System, KSAM
15 - File System, IMAGE
P4,P5,P6 Not used.

Event 84 (%124)

```
Event Name: INITIATE
Description: Called when starting/completing logical disc
             request.
Calling Module: CACHESEG
Calling Procedures: CDT'Initiator, CDT'Completor

Parameter Description
--------------------

P1      = CDT Mapped Domain entry number
P2      = LDR entry index
P3      = type
          0 = Initiator
          1 - Completor
P4,P5,P6 Not used.
```

Event 86 (%126)

```
Event Name: CDT_ATT
Description: Called from CDT'ATTACHIO.
Calling Module: CACHESEG
Calling Procedure: CDT'Attachio

Parameter Description
---------------------

P1      = Ldev
P2      = Function
P3      = Flags
P4,P5   = Parm1, Parm2
P6      = Count
```

Event 87 (%127)

```
Event Name: MAP_DOM
Description: Called when need to "map" a disc domain.
Calling Module: CACHESEG
Calling Procedure: CDT'MAP'CACHED'DOMAIN

Parameter Description
---------------------

P1      = New CDT entry number
P2      = Returned CDT entry
P3,P4,P5,P6 Not used.
```

Event 88 (%130)

```
Event Name: UN_MAP_RG
Description: Called when disc domain no longer mapped. (i.e. both
             the logical and physical I/O is complete).
Calling Module: CACHESEG
Calling Procedure: CDT'MAP'CACHED'REGION

Parameter Description
---------------------

P1      = CDT Ldev entry number
P2      = Region CDT entry number
P3,P4,P5,P6 Not used.
```

Event 89 (%131)

```
Event Name: LINK_REG
Description: Called when a disc domain gets linked into the
             linked list of domains for an ldev.
Calling Module: CACHESEG
Calling Procedure: LINK'CACHED'REGION,UNLINK'CACHED'REGION

Parameter Description
---------------------

P1      = Type
            0 = Link
            1 = Unlink
P2,P3   = Address of region base
P4      = CDT entry number found in the header
P5      = # of pages
P6  Not used.
```

MMSTAT Event Group 9 (Disc I/O Requests)

Event 90 (%132)

```
Event Name: REQCACHE
Description: Called to see if caching will accept this
             I/O request.
Calling Module: CACHESEG
Calling Procedure: REQUEST'CACHE

Parameter Description
----------------------

P1      = LDR entry index
P2,P3,P4,P5,P6 Not used.
```

Event -98 (%142)

```
EVENT NAME: DISK TRAFFIC
DESCRIPTION: DISC I/O REQUEST HAS BEEN QUEUED

  CALLING MODULE: HARDRES

      CALLING PROCEDURE: ATTACHIO

                 PARAMETERS      PARAMETER DESCRIPTION

                     P1=CNT      DATA TRANSFER COUNT:WORDS IF >0;
                                 BYTES IF <0
                     P2=FLAGS.(0:4)
                     P3=FNCT       =0 ==>READ
                                   =1 ==>WRITE
                                   =2 ==>OPEN FILE
                                   =3 ==>CLOSE FILE
                                   =4 ==>CLOSE DEVICE
```

MMSTAT Event Group 10

Event 100 (%144)

```
EVENT NAME: DISK ERROR
DESCRIPTION:  RECORD DISC ERROR

  CALLING MODULE:  IOFDISC1

      CALLING PROCEDURE: FHDDVR

                 PARAMETERS      PARAMETER DESCRIPTION

                     P1=DIPT(DSTAT)     HARDWARE STATUS
                     P2=SO               QMISC
                     P3=IOQP(QLDEV).QLDEVN LOR STOCOUNT&LSL(8))
                       =LDEV/SIO PROGRAM COUNTER
```

Event 101 (%145)

```
EVENT NAME: DISK ERROR
DESCRIPTION:  RECORD DISC ERROR

  CALLING MODULE: IOMDISC0

      CALLING PROCEDURE: MHDDVR

                 PARAMETERS      PARAMETER DESCRIPTION


                     P1=DIPT(DSTAT)     HARDWARE STATUS
                     P2=SO               QMISC
                     P3=IOQP(QLDEV).QLDEVN LOR STOCOUNT&LSL(8))
                       =LDEV/SIO PROGRAM COUNTER
```

MMSTAT Event Group 11

Event -110 (%156)

```
EVENT NAME: START I/O
DESCRIPTION:DRIVER INITIATOR FOR SIO DEVICE HAS BEEN CALLED

  CALLING MODULE: HARDRES

      CALLING PROCEDURE: SIODM

                 PARAMETERS      PARAMETER DESCRIPTION

                     P1=IOQPL(QSTAT) LOR IOQPL(QLDEV).LDEVN
                       =(0:8) PCB ENTRY # OF PROCESS MAKING REQUEST
                        (8:8) LOGICAL DEVICE NUMBER OF DEVICE FOR I/O
                     P2=IOQP(QUBCT)=WORD COUNT IF>0;BYTE COUNT IF<0
                     P3=(0:2) = FUNCTION CODE SPECIFIED BY DRIVER

                              = 0  => READ
                              = 1  => WRITE
                              = 2  => CONTROL

                        =(6:10)= DSTN OF TARGET DATA SEG
```

Event -111 (%157)

```
EVENT NAME: I/O COMPLETION
DESCRIPTION: SIO COMPLETION

  CALLING MODULE: HARDRES

      CALLING PROCEDURE: SIODM

                 PARAMETERS      PARAMETER DESCRIPTION

                     P1=IOQP(QLDEV).LDEVN=LOGICAL DEVICE NUMBER OF
                                          DISC INVOLVED IN TRANSFER
                     P2=IOQP(QPAR1)   (DEFINED BY DRIVER)
                     P3=IOQP(QPAR2)   (DEFINED BY DRIVER)
```

MMSTAT Event Group 12

Event 120 (%170)

```
EVENT NAME: SOFT'DEATH
DESCRIPTION: BUG CATCHER

  CALLING MODULE: HARDRES

      CALLING PROCEDURE: SOFT'DEATH

                 PARAMETERS      PARAMETER DESCRIPTION

                     P1          SOFT'DEATH I.D. NUMBER
                     P2          CALLERS STATUS REGISTER
                     P3          CALLERS DELTA P
```

Event 125 (%175)

```
EVENT NAME:  IOBUFTRP
EVENT DESCRIPTION:  IOSYSTEM BUFFER TRAP

CALLING MODULE:  HARDRES
CALLING PROCEDURE:  SIODM

     PARAMETER DESCRIPTION
       -----
     P1 = IOQP
     P2 = IOQP(QDSTN).DSTN = DST NUMBER OF BUFFER
     P3 = 0
```

MMSTAT Event Group 13

Event 139 (%213)

```
Event Name: C_ABSENT
Description: Either the mapped disc domain or the target
             DST was absent when a cache move was attempted.
Calling Module: CACHESEG
Calling Procedure: PROCESSCDTLOGREQQUEUE

Parameter Description
---------------------

P1     = 0  Mapped Domain absent
P2     = Pin
P3,P4  = Segment identifier of Mapped Domain
P5,P6 Not used.


P1     = LDR entry index  (DST not present)
P2     = Pin
P3,P4  = Segment identifier of DST (P4.(0:1) = 1 stack)
P5,P6 Not used.
```

MMSTAT Event Group 14 (CS/3000)

Event 140 (%214)

```
EVENT  NAME: COPEN
DESCRIPTION:

  CALLING MODULE: COMSYS2

      CALLING PROCEDURE: COPEN

                 PARAMETERS      PARAMETER DESCRIPTION

                     P1  (0:8) = CS ERROR CODE
                         (8:8) = LOGICAL DEVICE NUMBER

                     P2  PMAP1

                     P3  PMAP2
```

Event 142 (%216)

EVENT NAME: CABORTIO
DESCRIPTION:

CALLING MODULE: COMSYS1

CALLING PROCEDURE: CABORTIO

PARAMETERS	PARAMETER DESCRIPTION
P1	LOGICAL DEVICE
P2	IOQINDEX
P3	0

Event 144 (%220)

EVENT NAME: CSIOWAIT
DESCRIPTION:

CALLING MODULE: COMSYS1

CALLING PROCEDURE: CSIOWAIT

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	TRANSMISSION LOG
P3	

Event 146 (%222)

EVENT NAME: CCLOSE
DESCRIPTION:

CALLING MODULE: COMSYS3

CALLING PROCEDURE: CCLOSE

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	LINE NUMBER
P3	0

Event 147 (%223)

EVENT NAME: CREAD
DESCRIPTION:

CALLING MODULE: COMSYS4

CALLING PROCEDURE: CREAD

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	INCOUNT
P3	STATION

Event 149 (%225)

EVENT NAME: CWRITE
DESCRIPTION:

CALLING MODULE: COMSYS4

CALLING PROCEDURE: CWRITE

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	OUTCOUNT
P3	INCOUNT

MMSTAT Event Group 15 (CS/3000)

Event 150 (%226)

EVENT NAME: CSDRIVER
DESCRIPTION:

CALLING MODULE: BSCLCM

CALLING PROCEDURE: CSDRIVER

PARAMETERS	PARAMETER DESCRIPTION
P1 TIMER	LSW
P2 CURRENTSTATE	WHERE THE DRIVER IS IN THE STATE TRANSITION TABLE
P3 CURRENTEVENT	(0:8) = CURRENT EVENT (8:8) = LOGICAL DEVICE WHAT CAUSED THE DRIVER TO BECOME ACTIVE

Event 152 (%230)

EVENT NAME: CCONTROL
DESCRIPTION

CALLING MODULE: COMSYS5

CALLING PROCEDURE: CCONTROL

PARAMETERS	PARAMETER DESCRIPTION
P1	(0:8) = CS ERROR CODE (8:8) = LOGICAL DEVICE NUMBER
P2	CONTROL CODE
P3	PARAMETER

Event 153 (%231)

EVENT NAME: COPENTRACEFILE
DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE: COPENTRACEFILE

```
PARAMETERS      PARAMETER DESCRIPTION

    P1  (0:8) = CS ERROR CODE
        (8:8) = LOGICAL DEVICE NUMBER

    P2  CTRACEINFO

    P3  0
```

Event 154 (%232)

EVENT NAME: CCLOSETRACEFILE
DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE: CCLOSETRACEFILE

```
PARAMETERS      PARAMETER DESCRIPTION

    P1  (0:8) = CS ERROR CODE
        (8:8) = LOGICAL DEVICE NUMBER

    P2  0

    P3  0
```

Event 155 (%233)

EVENT NAME: CPOLLIST
DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE: CPOLLIST

```
PARAMETERS      PARAMETER DESCRIPTION

    P1  LOGICAL DEVICE


    P2  CS ERROR CODE

    P3  PMAP
```

MMSTAT Event Group 16

Event 160 (%240)

EVENT NAME: CREAD
DESCRIPTION:

CALLING MODULE: DSMON

CALLING PROCEDURE:

```
PARAMETERS      PARAMETER DESCRIPTION

    P1= TIME STAMP

    P2= (0:4) NOT USED
        (4:1) BLOCK
        (5:2) STATE
        (7:3) NEXT
        (10:1) :=0  INITIALIZATION EVENT
               :=1  COMPLETION EVENT
        (11:5) SUB EVENT NUMBER

    P3= DEPENDS ON THE SUB EVENT NUMBER AND
        IF IT IS AN INITIALIZATION OR COMPLETION EVENT.
        MSG: (0:4)   STRMTYPX
             (4:6)   MSG CLS
             (10:16) STRMTYP
```

SUB EVENT NO.	SUB EVENT NAME	INIT PARM	COMP PARM
0	CREAD	0	LEN
1	CWRITE	X MSG	LEN
2	IOWAIT	0	LEN
3	CCHECK	0	ERRCOD
4	DSATTN	0	0
5	DSWC	X MSG	R MSG
6	CHNGEWAIT	PARM	0
7	MONREQ	REQ	0
10	CABORT	0	T/F
11	CRESET	0	0
12	CSDATA	R MSG	
13	CSREREAD		

MMSTAT Event Group 19

Event 191 (%277)

EVENT NAME: DISKINTRPT
DESCRIPTION: A 7905/7920 CONTROLLER IS PROCESSING AN ATTENTION INTERRUPT (ONLINE/OFFLINE)
CALLING MODULE: HARDRES

CALLING PROCEDURE: SIODM

```
PARAMETERS      PARAMETER DESCRIPTION
    P1= @DITP     (US)--i.e. WHO GOT THE INTERRUPT


    P2= @DITP     (THEM)--i.e. WHO RAN THE POLL PROGRAM


    P3= DITP      "OUR" DIT FLAGS WORD
```

THERE SHOULD BE AT LEAST AN %300 AND AN %303 FOR EACH SIO PRGM. A SINGLE ISOLATED (IN TIME) REQUEST WILL GENERATE AT LEAST A %303, %300, %303. IF THE QUEUE OF IOQ'S ON A DIT NEVER EMPTIES, THERE WOULD BE ONE %300 AND ONE %303 PER SIO PRGM.

Event 192 (%300)

EVENT NAME: GIPINTERRUPT
DESCRIPTION: INTERRUPT JUST PROCESSED

CALLING MODULE: HARDRES

CALLING PROCEDURE: GIP

PARAMETERS		PARAMETER DESCRIPTION
P1	=	LDEV
P2	=	QUEUE ELEMENT WORD ENTRY INDEX
P3	=	CONTENTS OF DIT WORD 0: THE FLAGS WORD
P4	=	CHANNEL PROGRAM INSTRUCTION POINTER
P5	=	CONTROLLER STATUS
P6	=	LSW of a Return from TIMER

Event 193 (%301)

EVENT NAME: STARTIO
DESCRIPTION: Issuing SIOP machine instruction.

CALLING MODULE: HARDRES

CALLING PROCEDURE: START'HPIB, STARTIO

PARAMETERS		PARAMETER DESCRIPTION
P1	=	Absolute address of SIO program to start.
P2	=	LDEV number
P3	=	DRT number
P4	=	Q'ENTRY'INDEX FROM DITP(DIOQP)
P5	=	DIT WORD 0: THE DIT FLAGS WORD
P6	=	LSW of A RETURN FROM A CALL TO TIMER

Event 194 (%302)

EVENT NAME: SIODM-ENTRY
DESCRIPTION: Entering SIODM

CALLING MODULE: HARDRES

CALLING PROCEDURE: SIODM

PARAMETERS		PARAMETER DESCRIPTION
P1	=	LDEV
P2	=	IOQ OR DRQ table relative index
P3	=	DIT WORD 0 (DIT FLAGS)
P4	=	CURRENT STATE OF THE VARIABLE STATE IN SIODM
P5	=	UNUSED AT THIS TIME
P6	=	LSW RETURNED BY CALL TO TIMER

Event 195 (%303)

EVENT NAME: SIODM-EXIT
DESCRIPTION: Leaving SIODM main loop.

CALLING MODULE: HARDRES

CALLING PROCEDURE: SIODM

PARAMETERS PARAMETER DESCRIPTION

SAME AS EVENT 194 (%302)
EXCEPT THAT EVENT IS 195 (%303)

MMSTAT Event Group 20

THESE EVENTS ARE FOR DEVELOPMENT USE ONLY AND ARE NOT NORMALLY ENABLED

Event 200 (%310)

EVENT NAME: DISKBUGCATCHER
DESCRIPTION: A MOUNTED VOLUME TABLE CHANGE IS BEING MADE.

CALLING MODULE: PVSYS

CALLING PROCEDURE: MVTABLE

PARAMETERS PARAMETER DESCRIPTION

P1= FUNCT
0 = DELETE ENTRY
1 = ADD ENTRY
2 = PRESERVE ENTRY

P2= MVTABX (MOUNTED VOLUME TABLE INDEX)

P3= DELTAP (VALUE OF Q-2)

Event 201 (%311)

EVENT NAME: DISKBUGCATCHER
DESCRIPTION: A PRIVATE VOLUME USER TABLE CHANGE IS BEING MADE.

CALLING MODULE: PVSYS

CALLING PROCEDURE: USERTABLE

PARAMETERS PARAMETER DESCRIPTION

P1= FUNCT
0 = CREATE USER ENTRY
1 = RENAME USER ENTRY
2 = RETURN ALL MVTABX INDICES USED BY A SPECIFIC PCB
3 = RETURN ALL PCB POINTERS USING A SPECIFIC MVTABX
4 = GET USER ENTRY

P2= MVTABX (MOUNTED VOLUME TABLE INDEX)

P3= DELTAP (VALUE OF Q-2)

MMSTAT Event Group 21 Process Creations and Terminations Logical Process Table

Event -211 (%323)

EVENT NAME: PROCESS COMPLETION
DESCRIPTION: PROCESS HAS TERMINATED

CALLING MODULE: MORGUE

CALLING PROCEDURE: TERMINATE

```
PARAMETERS      PARAMETER DESCRIPTION

    P1=0
    P2=0
    P3=0
```

MMSTAT Event Group 22

Time Stamp of Event Trace Enable and Disable

Event 221 (%335)

EVENT NAME: CONFIGURATION INFORMATION
DESCRIPTION: EVENT GROUP MASK

CALLING MODULE: CRIO

CALLING PROCEDURE: CONSMON

```
PARAMETERS      PARAMETER DESCRIPTION

    P1= MEASMSK0

    P2= MEASMSK1

    P3=Reserved
```

Event 222 (%336)

EVENT NAME: CONFIGURATION INFORMATION
DESCRIPTION: MPE VERSION FIX UPDATE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

```
PARAMETERS      PARAMETER DESCRIPTION

    P1= VERSION


    P2= FIXL


    P3= UPDATEL
```

Event -223 (-%337)

EVENT NAME: CONFIGURATION INFORMATION
DESCRIPTION: SYSTEM TABLE LOCATIONS AND AVAILABLE LINKED MEMORY
INFORMATION
CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

```
PARAMETERS      PARAMETER DESCRIPTION

    P1=F (%1032)=@CST(0)-@DST(0)
                =DISPLACEMENT TO CODE
    P2=F(%1033)=@CST(LAST)-@DST(0)
                =DISPLACEMENT TO SHARABLE
    P3=LOGICAL(TOTAL&DLSK(4))=LINKED MEMORY SIZE
```

Event -224 -(%340)

EVENT NAME: SYSPINS
DESCRIPTION: LOGICAL PROCESS TABLE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

```
PARAMETERS      PARAMETER DESCRIPTION

    P1=ABSOLUTE(%1141)=PROGEN'S PCBENTRY NUMBER
    P2=ABSOLUTE(%1142)=MAM'S PCB ENTRY NUMBER
    P3=ABSOLUTE(%1143)=UCOP'S PCB ENTRY NUMBER
```

Event -225 (-%341)

EVENT NAME: SYSPINS(CNTD.)
DESCRIPTION: LOGICAL PROCESS TABLE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

```
PARAMETERS      PARAMETER DESCRIPTION

    P1=ABSOLUTE(%1144)=PFAIL'S PCB ENTRY NUMBER
    P2=ABSOLUTE(%1145)=DEVREC'S PCB ENTRY #
    P3=ABSOLUTE(%1146)=PRMSG'S PCB ENTRY #
```

Event -226 (-%342)

EVENT NAME: SYSPINS(CNTD.)
DESCRIPTION: LOGICAL PROCESS TABLE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE: CXMON

```
PARAMETERS      PARAMETER DESCRIPTION

    P1=ABSOLUTE(%1147)=STMSG'S PCB ENTRY #
    P2=ABSOLUTE(%1150)=LOG'S PCB ENTRY #
    P3=ABSOLUTE(%1151)=LOAD'S PCB ENTRY #
```

Event -227 (-%343)

```
EVENT  NAME: SYSPINS(CNTD.)
DESCRIPTION: LOGICAL PROCESS TABLE

  CALLING MODULE: OPCOMMAND

      CALLING PROCEDURE: CXMON

                 PARAMETERS      PARAMETER DESCRIPTION

                     P1=ABSOLUTE(%1152)=IOMESSPROC'S PCB ENTRY #
                     P2=ABSOLUTE(%1153)=SYSIOPROC'S PCB ENTRY #
                     P3=ABSOLUTE(%1154)=MEMLOGP'S PCB ENTRY #
```

Event -228 (%344)

```
EVENT  NAME: TIMESTAMP
DESCRIPTION: TIMESTAMP

  CALLING MODULE: OPCOMMAND

      CALLING PROCEDURE: CXMON

                 PARAMETERS      PARAMETER DESCRIPTION

                     P1=CALENDAR     (0:7)=YEAR OF CENTURY
                                     (7:9)=DAY OF YEAR
                     P2=CLOCK(WORD1).(0:7)=HOUR OF DAY
                                     (8:8)=MINUTE OF HOUR
                     P3=CLOCK(WORD2).(0:7)=SECONDS INTO MINUTE
                                    .(8:8)=TENTHS OF SECONDS
```

Event -229 (-%345)

```
EVENT  NAME: MONOFF
DESCRIPTION: END EVENT TRACING

  CALLING MODULE: OPCOMMAND

      CALLING PROCEDURE: CXMON

                 PARAMETERS      PARAMETER DESCRIPTION

                     P1=0
                     P2=0
                     P3=0
```

MMSTAT Event Group 23 (Terminal I/O)

Event 230 (%346)

```
EVENT NAME:  TERMREAD
DESCRIPTION:  TERMINAL READ COMPLETION

     CALLING MODULE:  HARDRES
     CALLING PROCEDURE:  TIP

          PARAMETERS          PARAMETER DESCRIPTION

          P1 = LDEV
          P2 = READ DURATION
          P3 = BYTES READ
```

Event 231 (%347)

```
EVENT NAME:  DC1DC2ACK
DESCRIPTION:  DC1/DC2 HAS BEEN SATISFIED

     CALLING MODULE:  HARDRES
     CALLING PROCEDURE:  TIP

     PARAMETERS          PARAMETER DESCRIPTION

     P1 = LDEV
     P2 = DURATION (BETWEEN START AND DC2)
     P3 = BYTES READ (EXCLUDING DC2)
```

Event 232 (%350)

```
EVENT NAME:  TERMWRITE
DESCRIPTION:  WRITE COMPLETION

     CALLING MODULE:  IOTERMO
     CALLING PROCEDURE:  TERMIOM

     PARAMETERS          PARAMETER DESCRIPTION

     P1 = LDEV
     P2 = 0
     P3 = BYTE COUNT OF TRANSFER
```

Event 233 (%351)

```
EVENT NAME:  BINREAD
DESCRIPTION:  BINARY READ COMPLETED

     CALLING MODULE:  HARDRES
     CALLING PROCEDURE:  TIP

     PARAMETERS          PARAMETER DESCRIPTION
     P1 = LDEV
     P2 = DURATION
     P3 = BYTES READ
```

Event 234 (%352)

```
EVENT NAME:  TERMLOGON
DESCRIPTION:  TERMINAL JUST LOGGING ON

     CALLING MODULE:  IOTERMO
     CALLING PROCEDURE:  TERMIOM

     PARAMETERS          PARAMETER DESCRIPTION
     P1 = LDEV
     P2 = 0
     P3 = 0
```

Event 235 (%353)

```
EVENT NAME:  TERMLOGOFF
DESCRIPTION:  TERMINAL JUST LOGGED OFF

     CALLING MODULE:  IOTERMO
     CALLING PROCEDURE:  TERMIOM

     PARAMETERS          PARAMETER DESCRIPTION
     P1 = LDEV
     P2 = 0
     P3 = 0
```

Event 236 (%354)

EVENT NAME: SPECCHAR
DESCRIPTION: PROCESSED SPECIAL CHARACTER

```
     CALLING MODULE:  HARDRES
     CALLING PROCEDURE:  TIP

     PARAMETERS          PARAMETER DESCRIPTION
     P1 = LDEV
     P2 = SPECIAL CHARACTER PROCESSED
     P3 = 0
```

Event 237 (%355)

EVENT NAME: BREAK
DESCRIPTION: PROCESSED BREAK

```
     CALLING MODULE:  HARDRES
     CALLING PROCEDURE:  TIP

     PARAMETERS          PARAMETER DESCRIPTION
     P1 = LDEV
     P2 = DSTATE
     P3 = 0
```

Event 238 (%356)

EVENT NAME: SPECREAD
DESCRIPTION: SPECIAL READ TERMINATION CHARACTER DETECTED

```
     CALLING MODULE:  HARDRES
     CALLING PROCEDURE:  TIP

     PARAMETERS          PARAMETER DESCRIPTION

     P1 = LDEV
     P2 = DURATION
     P3 = BCNT
```

MMSTAT Event Group 24 (Power Fail)

Event 240 (%360)

Event Name: PFAIL
Description: Power fail detected.
Calling Module: ININ, PFAIL
Calling Procedures: Powerup (ININ), Powerup (PFAIL)

```
Parameter Description
---------------------

P1  =  0   Called from Powerup in ININ
       1   Called from entry in Powerup in PFAIL
       2   Called from end of Powerup in PFAIL

P2  = For P1=0 this is 0
      For P1=1,2:
      TRUE = Multiple powerfail
      FALSE= First powerfail

P3  = PF
      0  = No powerfail or PFAIL processing complete
      1  = Set by the power down trap in ININ
      2  = Set by the power up trap in ININ
      3  = Set when awake the PFAIL process
      4  = Set by PFAIL after message appears on console

P4  = SYSUP
      0  = System not back up after powerfail
      1  = System back up after powerfail

P5,P6 not used.
```

CHAPTER 21 ROOTFILE LAYOUT

General Rootfile Layout

```
            ________________________
           |                        |
  LABEL 0  | ROOTFILE INFORMATION   |
           |________________128_wds|
            ________________________
           |                        |
        1  | PASSWORD TABLE         |
           |________________________|
            ________________________
           |                        |
        2  | PASSWORD TABLE (CONT.) |
           |________________________|
            ________________________
           |                        |
        3  | ITEM R/W TABLE         |
           |________________________|
           .                        .
           .                        .
           .________________________.
           |                        |
        .  | SET R/W TABLE          |
           |________________________|

            ________________________
           |                        |
 RECORD 0  | DATABASE GLOBAL INFO   |
           |________________128 wds|
            ________________________
           |                        |
        1  | ITEM TABLE             |
        .  | (variable size)        |
        .  |------------------------|
        .  | SET TABLE              |
        .  | (variable size)        |
        .  |------------------------|
        .  | DATA SET CONTROL BLOCKS|
        .  |                        |
        .  |         (DSCB)         |
        .  |                        |
        .  |    (variable size)     |
        .  |________________________|
```

The data base ROOT FILE is an MPE file with filecode equal to -400. The record size is 128 words, fixed, binary format with a blocking factor of 1. The size of the file depends on the number of data items and data sets defined in the data base.

Root File Label 0

```
          _________________________________________   %
  WORD 0 |_RL'CONDITION____(rootfile_condition)___|  0
       1 |_RL'DATE_________(creation_date)________|  1
       2 | RL'TIME         (creation time)        |  2
       3 |________________________________________|  3
       4 |_RL'EVEROPEN____________________________|  4
       5 |_RL'COLDLOADID___(cold_load_id)_________|  5
       6 |_RL'USERCOUNT___________________________|  6
       7 |_RL'DBCBDSTNUM___(DST_number_of_DBCB)___|  7
       8 | RL'LOGID        (log id for            | 10
       . |                  transaction logging)  |  .
       . |                                        |  .
      11 |________________________________________| 13
      12 | RL'LOGPASS      (log id password)      | 14
       . |                                        |  .
       . |                                        |  .
      15 |________________________________________| 17
      16 |_RL'FLAGS________(database_flags)_______| 20
      17 |_RL'STORDATE_____(DBSTORE_date)_________| 21
      18 | RL'STORTIME     (DBSTORE time)         | 22
      19 |________________________________________| 23
      20 |_RL'BUFSPECCOUNT__(buffer_spec_count)___| 24
      21 |_RL'ILRCREATEDATE_(date_ILR_log_created)_| 25
      22 | RL'ILRCREATETIME (time ILR log created) | 26
      23 |________________________________________| 27
      24 |_RL'ILRLASTDATE___(last_log_access_date)_| 30
      25 | RL'ILRLASTTIME   (last log access time) | 31
      26 |________________________________________| 32
      27 | RESERVED                               | 33
      . .               FOR                       . .
      . .                         FUTURE          . .
      63 |____________________________________USE_| 77
      64 | RL'MAINTWORD     (database maintenance | 100
       . |                   word)                |  .
       . |                                        |  .
      67 |________________________________________| 103
      68 | RL'BUFFERSPECS   (buffer specifications)| 104
         |                                        |  .
      to |                                        |  .
         |                                        |  .
     127 |________________________________________| 177
```

RL'CONDITION (IN ASCII):

```
        JB - Virgin. The database has not been created yet.
        FW - OK. The database is OK.
        RM - Modified deferred. The database is being modified.
        MC - Maintenance create. The database is being created.
        ME - Maintenance erase. The database is being erased.
        IL - ILR recovery in progress.
```

Root File Label 0 (cont.)

RL'DATE: Root file creation date*. Its format is:

```
 _0:_1:_2:_3:_4:_5:_6:_7:_8:_9:10:11:12:13:14:15
|year______________|day_of_year______________|
```

RL'TIME: Root file creation time*. Its format is:

```
 _0:_1:_2:_3:_4:_5:_6:_7:_8:_9:10:11:12:13:14:15
|hour________________|minutes_______________|
|seconds_____________|tenth_of_seconds______|
```

RL'EVEROPEN: This field is no longer used under IMAGE B

RL'FLAGS:

```
        (0:1) - RECOVERY         Default is NO  (0)
        (1:1) - LOGGING          Default is NO  (0)
        (2:1) - ACCESS           Default is YES (1)
        (3:1) - DUMPING          Default is NO  (0)
        (4:1) - RESERVED-FOR-FUTURE-USE
        (5:2) - SUBSYSTEM ACCESS Default is R/W (00)
        (7:1) - ILR              Default is NO  (0)
        (8:2) - RESERVED-FOR-FUTURE-USE
        (10:1)- DIRTY FLAG       Default is YES (1).
                                 This indicates the database has
                                 been modified but not DBSTOREd.
        (11:5)- RESERVED-FOR-FUTURE-USE
```

RL'STORDATE: Same format as RL'DATE*.

RL'STORTIME: Same format as RL'TIME*.

RL'BUFSPECCOUNT: Maximum number of buffer specifications allowed.

RL'ILRCREATEDATE: Same format as RL'DATE*.

RL'ILRCREATETIME: Same format as RL'TIME*.

RL'ILRLASTDATE: Same format as RL'DATE*.

RL'ILRLASTTIME: Same format as RL'TIME*.

RL'MAINTWORD: For data bases with no maintenance word this field has 2 semicolons (';;') and trailing blanks.

RL'BUFFSPECS:

```
  BIT/    _0:_1:_2:_3:_4:_5:_6:_7:_8:_9:10:11:12:13:14:15   %
  WD 68 |buffers_for_1 user____|buffers_for_2 users____| 104
     69 |buffers_for_3 users___|buffers_for_4 users____| 105
      . | etc...                                        |  .
      . |_______________________________________________|  .
    127 |buffers_for_119 users_|buffers_for_120 users__| 177
```

* The DATE and TIME fields can be formatted (for display purposes) individually by calling the FMTCALENDAR and FMTCLOCK Intrinsics respectively. Or both fields can be formatted at once with FMTDATE Intrinsic.

Root File Labels 1 & 2

```
         _LABEL_#1_____________________________   %
  WORD 0 | Password for user class 0           | 0
       1 | (this is a dummy field since user   | 1
       2 |  class 0 is not defined)            | 2
       3 |_____________________________________| 3
       4 | Password for user class 1           | 4
       5 |                                     | 5
       6 |                                     | 6
       7 |_____________________________________| 7
       8 | Password for user class 2           | 10
       9 |                                     | 11
      10 |                                     | 12
      11 |_____________________________________| 13
       . |                                     | .
       . .                                     . .
       . |_____________________________________| .
     124 | Password for user class 31          | 174
     125 |                                     | 175
     126 |                                     | 176
     127 |_____________________________________| 177

         _LABEL_#2_____________________________   %
       0 | Password for user class 32          | 0
       1 |                                     | 1
       2 |                                     | 2
       3 |_____________________________________| 3
       4 | Password for user class 33          | 4
       5 |                                     | 5
       6 |                                     | 6
       7 |_____________________________________| 7
       8 | Password for user class 34          | 10
       9 |                                     | 11
      10 |                                     | 12
      11 |_____________________________________| 13
       . |                                     | .
       . .                                     . .
       . |_____________________________________| .
     124 | Password for user class 63          | 174
     125 |                                     | 175
     126 |                                     | 176
     127 |_____________________________________| 177
```

The PASSWORD TABLE occupies user labels number 1 and 2.
There are four words (8 characters) reserved for each password. The relative position of a password corresponds to the user class number defined in the schema. For user class numbers not defined in the SCHEMA, the four word field is filled with blanks.

Root File Label 3

```
         _LABEL_#3_____________________________   %
  WORD 0 | Item1 read/write bit map            | 0
       1 |                                     | 1
       2 |                                     | 2
       3 |                                     | 3
       4 |                                     | 4
       5 |                                     | 5
       6 |                                     | 6
       7 |_____________________________________| 7
       8 | Item2 read/write bit map            | 10
       9 |                                     | 11
       . :                                     : .
       . :                                     : .
       . :                                     : .
      15 |_____________________________________| 17
      16 | Item3 read/write bit map            | 20
      17 |                                     | 21
       . :                                     : .
       . :                                     : .
       . :                                     : .
     119 |_____________________________________| 167
     120 | Item16 read/write bit map           | 170
     121 |                                     | 171
       . :                                     : .
       . :                                     : .
       . :                                     : .
     127 |_____________________________________| 177
```

The ITEM READ/WRITE TABLE starts in user label #3
There are eight words for each ITEM READ/WRITE bit map.
For databases with more than 16 items, the read/write table continues in the next user labels. The specific format of this table is explained after the SET READ/WRITE TABLE since it is defined the same way.
The number of user labels occupied by the ITEM READ/WRITE TABLE depends on the number of data items defined in the schema and can be obtained by rounding upwards (ceiling)the result of:

Num-of-labels = [(Num-of-items)*8]/128

Since there can only be a maximum of 255 data items in the schema, the maximum size for this table in user labels would be:

Max-size = [(255)*8]/128 = 15.93 => 16 labels.

Root File- Next Label

```
         _LABEL_#?_____________________________   %
  WORD 0 | Set1 read/write bit map             | 0
       1 |                                     | 1
       2 |                                     | 2
       3 |                                     | 3
       4 |                                     | 4
       5 |                                     | 5
       6 |                                     | 6
       7 |_____________________________________| 7
       8 | Set2 read/write bit map             | 10
       9 |                                     | 11
       . :                                     : .
       . :                                     : .
       . :                                     : .
      15 |_____________________________________| 17
      16 | Set3 read/write bit map             | 20
      17 |                                     | 21
       . :                                     : .
       . :                                     : .
       . :                                     : .
     119 |_____________________________________| 167
     120 | Set16 read/write bit map            | 170
     121 |                                     | 171
       . :                                     : .
       . :                                     : .
       . :                                     : .
     127 |_____________________________________| 177
```

The SET READ/WRITE TABLE starts on a user label boundary after the ITEM READ/WRITE TABLE.
There are eight words for each SET READ/WRITE bit map.
For databases with more than 16 data sets, the read/write table continues in the next user labels.
The specific format of this table is shown in the next page.

The number of user labels occupied by the SET READ/WRITE TABLE depends in the number of data sets defined in the schema, and is obtained by rounding upwards (ceiling) the result of:

Num-of-labels = [(Num-of-sets)*8]/128

Since there can only be a maximum of 99 data sets defined in the schema the maximum size for this table in user labels is:

Max-size = [(99)*8]/128 = 6.18 => 7 labels

Item/Set Read/Write Table Format

There are eight words per item/set read/write table definition and up to 16 items/sets per record (user label). Within each 8 words, the first 4 words are the flags for the user classes which have read access to the item/set. The second 4 words are the flags for the user classes which have write access to the item/set. The detail format for an eight word field is shown below.

A. Four words for read access:

```
 0___________15_16__________31_32__________47_48__________63
|_word_1________|_word_2________|_word_3________|_word_4________|
```

4 words represent 64 bits. Bit n represents read access for user class n to the item/set. If bit n is set to 1 then user class n has read access to the item/set.
For example, if the word settings are:

```
  word 1   word 2   word 3   word 4
 %000016  %020000  %000410  %001300
```

This means that user classes 12, 13, 14, 18, 39, 44, 54, 56 and 57 have read access to the item/set.
If no read/write security is defined at all for the item/set, then all of the read security bits are set to 1.

B. Four words for write access:

```
 0___________15_16__________31_32__________47_48__________63
|_word_1________|_word_2________|_word_3________|_word_4________|
```

Write access flags have the same format as the read access flags. Bit n represents write access for user class n to the item/set. If bit n is set to 1, then user class n has write access to the item/set
For example, if the word settings are:

```
  word 1   word 2   word 3   word 4
 %000010  %020000  %000000  %001100
```

This means that the user classes 12, 18, 54 and 57 have write access to the item/set.
If no read/write security is defined at all for the item/set, then all of the write security bits are set to 0.

Root File Record 0

```
            _RECORD_#0_________________________________  %
   word 0 |_ROOT'DBSTATUS______________________________| 0
        1 | ROOT'DBNAME                                | 1
        2 |                                            | 2
        3 |                                            | 3
        4 |____________________________________________| 4
        5 |_ROOT'TRLRLGTH__(trailer_area_length)_______| 5
        6 |_ROOT'BUFFLGTH__(buffer_length)_____________| 6
        7 |_ROOT'LGTH______(rootfile_length)___________| 7
        8 |_ROOT'ITEMCT____(number_of_items)___________| 10
        9 |_ROOT'SETCT_____(number_of_data_sets)_______| 11
       10 |_ROOT'ITEMPTR___(item_table_pointer)________| 12
       11 |_ROOT'DSETPTR___(set_table_pointer)_________| 13
       12 | RESERVED       (set to blanks)             | 14
       13 |                                            | 15
       14 |                                            | 16
       15 |____________________________________________| 17
       16 |_NOWOPEN____________________________________| 20
       17 |_MAXOPEN____________________________________| 21
       18 | RESERVED       (for future use)            | 22
        . :                (set to binary 0s)          : .
        . :                                            : .
        . :                                            : .
      127 |____________________________________________| 177
```

ROOT'DBSTATUS
(0:8) - IMAGE version ('B' in ASCII)
(8:8) - Binary 1 (filler)

ROOT'DBNAME - DATABASE name left justified (last 2 chars are blank).

NOWOPEN - Number of data sets opened. This field is not used in IMAGE B

MAXOPEN - Maximum number of data sets that can be opened. This field is not used in IMAGE B.

NOTE:
ROOT'ITEMPTR and ROOT'DSETPTR is a word offset from record 0 (beginning of the file, not including the space taken by the user labels) and can span several records.
These pointers point to the 0th entry of the table and since the 0th entry in the item table or the set table does not really exist, they actually point to 11 words before the beginning of the table. To get to the first entry in the table, this pointer should be incremented by the length of the entry (which is currently 11 words).

Root File Record 1

```
  bits/   _0:_1:_2:_3:_4:_5:_6:_7:_8:_9:10:11:12:13:14:15  %
  word 0 | item-name-1                                    | 0
       1 |                                                | 1
       2 |                                                | 2
       3 |                                                | 3
       4 |                                                | 4
       5 |                                                | 5
       6 |                                                | 6
       7 |________________________________________________| 7
       8 |_item-no-of-synonym____|_reserved-1_____________| 10
       9 |_reserved-2____________|_item-type______________| 11
      10 |_subitem-count_________|_subitem-length_________| 12
      11 | item-name-2                                    | 13
      12 |                                                | 14
      13 |                                                | 15
      14 |                                                | 16
      15 |                                                | 17
      16 |                                                | 20
      17 |                                                | 21
      18 |________________________________________________| 22
      19 |_item-no-of-synonym____|_reserved-1_____________| 23
      20 |_reserved-2____________|_item-type______________| 24
      21 |_subitem-count_________|_subitem-length_________| 25
      22 |                                                | 26
       . :                                                :
```

The ITEM TABLE starts in record #1.
Each entry is 11 words long and the length of the table depends on the number of data items defined in the schema. The relative position of an item definition depends on its relative position in the schema.

Item-name: is a data item name, left-justified and with trailing blanks

Item-number-of-synonym: is the number of the item whose name has the same hashed result as this one (this is utilized for quick item name searches).

Item-type: is one of the following: I, J, K, R, X, U, Z, or P

```
              item-type
              |
      VALUES, 20J2;
              | |subitem-length
              |subitem-count
```

The maximum size for this table is 11*255 = 2805wds

NOTES:
The reserved-1 and reserved-2 fields are the 'old' level numbers for read and write security. Now, the values are always zero.

Root File- Next Record

```
  bits/   _0:_1:_2:_3:_4:_5:_6:_7:_8:_9:10:11:12:13:14:15  %
  word 0 | set-name-1                                     | 0
       1 |                                                | 1
       2 |                                                | 2
       3 |                                                | 3
       4 |                                                | 4
       5 |                                                | 5
       6 |                                                | 6
       7 |________________________________________________| 7
       8 |_set-no-of-synonym_____|_reserved-1_____________| 10
       9 |_reserved-2____________|_data-set-type__________| 11
      10 |_DSCB-pointer___________________________________| 12
      11 | set-name-2                                     | 13
      12 |                                                | 14
      13 |                                                | 15
      14 |                                                | 16
      15 |                                                | 17
      16 |                                                | 20
      17 |                                                | 21
      18 |________________________________________________| 22
      19 |_set-no-of-synonym_____|_reserved-1_____________| 23
      20 |_reserved-2____________|_data-set-type__________| 24
      21 |_DSCB-pointer___________________________________| 25
      22 |                                                | 26
       . :                                                :
       . :                                                :
       . :                                                :
```

Set table follows the Item table.

Each entry is 11 words long. The length of the table depends on the number of data sets defined in the schema. The relative position of a set definition depends on its relative position in the schema.

Set-name: is a data set name, left-justified and with trailing blanks.

Set-number-of-synonym: is the number of a data set whose name has the same hashed result as this one (this is utilized for quick set name searches).

Data-set-type is one of the following: A, M or D.

DSCB-pointer: is a pointer to the Data Set Control Block. This pointer is word offset from record #0. The DSCB is described ahead.

The maximum size for this table is 11*99 = 1089wds.

NOTES: The reserved-1 and reserved-2 fields are the 'old' level numbers for the read and write access respectively. Since this concept no longer applies, the values are set to zero.

Data Set Control Blocks (DSCB)- General Layout

```
 ______________________________________________ _
| DATA SET GLOBAL AREA (set 1)                 | }
|      (capacity, lengths, counts, etc.)       | }
|                                     30 wds.  | }
|                                              | }
| RECORD DEFINITION TABLE (set 1)- - - - - - - | }
|         a. ITEM NUMBERS                      | } DSCB
|         b. ITEM DISPLACEMENT                 | } set1
|                             fieldcount*2+2   | }
|                                              | }
| PATH TABLE (set 1)- - - - - - - - - - - - - -| }
|      (search item, sort item, etc.)          | }
|                                pathcount*2   | }
|______________________________________________|_}
| DATA SET GLOBAL AREA (set 2)                 |  }
|      (capacity, lengths, counts, etc)        |  }
|                                     30 wds.  |  }
|                                              |  }
| RECORD DEFINITION TABLE (set 2)- - - - - - - |  }
|         a. ITEM NUMBERS                      |  } DSCB
|         b. ITEM DISPLACEMENT                 |  } set2
|                             fieldcount*2+2   |  }
|                                              |  }
| PATH TABLE (set 2)- - - - - - - - - - - - - -|  }
|      (search item, sort item, etc.)          |  }
|                                pathcount*2   |  }
|______________________________________________|__}
|                                              |
:                                              :
:                                              :
:                                              :
|______________________________________________|-
| DATA SET GLOBAL AREA (last set)              | }
|      (capacity, lengths, counts, etc)        | }
|                                     30 wds.  | }
|                                              | }
| RECORD DEFINITION TABLE (last set)- - - - - -| }
|         a. ITEM NUMBERS                      | } DSCB
|         b. ITEM DISPLACEMENT                 | } last set
|                             fieldcount*2+2   | }
|                                              | }
| PATH TABLE (last set)- - - - - - - - - - - - | }
|      (search item, sort item, etc.)          | }
|                                pathcount*2   | }
|______________________________________________|_}
```

The DSCBs follow the SET TABLE in the Root File.
There is one DSCB for each data set defined. The function of the DSCB is to define each data set within the data base.

Data Set Control Block (Global Area)

```
bit/    _0:_1:_2:_3:_4:_5:_6:_7:_8:_9:10:11:12:13:14:15  %
word 0 | DSCAP          (data set capacity)              | 0
     1 |_________________________________________________| 1
     2 |_DSBLOCKLGTH___(block_length)____________________| 2
     3 |_DSMEDIALGTH___(media_record_length)_____________| 3
     4 |_DSENTRYLGTH___(entry_length)____________________| 4
     5 |_DSBLOCKFAC______________|_DSFIELDCT_____________| 5
     6 |_DSPATHCT________________|_X|_DSPRIMKEY__________| 6
     7 |_DSPATHPTR_____(offset to path table)____________| 7
     8 | logical end of file                             | 10
     9 |_________________________________________________| 11
    10 | max num of records in set                       | 12
    11 |_________________________________________________| 13
    12 | 18 words of binary zeros                        | 14
     . :                                                 : .
     . :                                                 : .
     . :                                                 : .
    29 |_________________________________________________| 35
```

DSCAP - data set capacity as reported by the SCHEMA processor.

DSBLOCKLGTH - data set block length including the bit map overhead.

DSMEDIALGTH - data set media record length (remember that this length includes the pointer overhead)

DSENTRYLGTH - data set entry length.

DSBLOCKFAC - data set blocking factor.

DSFIELDCT - data set field count. This is the number of fields specified for the data set.

DSPATHCT - data set path count. This is the number of paths that are specified for the data set.

X-DSKEYTYPE - data set key type. If DSKEYTYPE = TRUE then the key is hashed.

DSPRIMKEY - data set primary path or key.
For master data sets, this is the field number of the search item.
For detail data sets, this is the field number of the primary path.

DSPATHPTR - data set path table pointer. Word offset to the data set path table which contains an entry for each path defined. It points to path 0th entry in the table, so to get to the first entry the pointer should be incremented by the length of the entry (which is currently 2 words).

Data Set Control Block (Item Numbers)

```
        _0:_1:_2:_3:_4:_5:_6:_7:_8:_9:10:11:12:13:14:15
word 0 |_item_num_of_1st_field_|_item_num_of_2nd_field_|
     1 |_item_num_of_3rd_field_|_etc.__________________|
     . |_etc.__________________|_binary 0______________|
     . |_binary_0______________|_binary_0______________|
```

The Item Numbers Table follows the Global Area of the DSCB. The size of this table (in words) is equal to the number of items in the given data set plus 1. The first n bytes are used to carry the item numbers of the fields within the data set. The remaining n+2 bytes are set to binary zeros.

Data Set Control Block (Record Definition Item Displacement)

```
        _0:_1:_2:_3:_4:_5:_6:_7:_8:_9:10:11:12:13:14:15
word 0 |_word_offset_to_first_field____________________|
     1 |_word_offset_to_second_field___________________|
     2 |_word_offset_to_third_field____________________|
     . :                                               :
     . :                                               :
     . :                                               :
     . |_word_offset_to_last_field_____________________|
       |_length_of_entry_______________________________|
```

This table immediately follows the Item Numbers Table.

The word offset points to the starting location of the field within the media record. Remember that the media record includes the pointer overhead so this offset varies for master and detail data sets: if a master data set has only one path, the word offset for the first field is 10, since there are 10 words of overhead--5 words for the synonym chain pointers and 5 words for the data set chain head that it would be pointing to. On a detail data set with one path, the overhead is only 4 words.

The 'length-of-entry' field is the same as the media record length.

Data Set Control Block (Path Table)

```
        _0:_1:_2:_3:_4:_5:_6:_7:_8:_9:10:11:12:13:14:15
word 0 | 1st path definition                           |
     1 |_______________________________________________|
     2 | 2nd path definition                           |
     3 |_______________________________________________|
     4 |                                               |
     . :                                               :
     . :                                               :
     . :_______________________________________________:
     . | last path definition                          |
     . |_______________________________________________|
```

There are 2 words (4 bytes) for each path definition.
The PATH TABLE for master data sets has a different layout from the PATH TABLE for detail data sets.

Master sets:
Byte Description
1 - item number of the search item in the related detail set.
2 - item number of the sort item in the related detail set.
3 - set number of the related detail data set
4 - path number of the corresponding path in the related detail data set.

Detail sets:
Byte Description
1 - field number of the search item.
2 - field number of the sort item.
3 - set number of the related master data set
4 - path number of the corresponding path in the related master data set.

General Data Set Layout

```
             __USER_LABEL_O_______________
Word  0-1|  masters=capacity             |
         |  details=highwater mark       |
         |_______________________________|
Word  2-3|                               |
         |  number of unused records     |
         |_______________________________|
Word  4-5|  masters= not used            |
         |details= delete chain head     |
         |_______________________________|
```

```
             __RECORD 0 through n________
Record  0   |                            |
            |                            |
            |     data records           |
            .                            .
            .                            .
            |                            |
Record  n   |____________________________|
```

Data Set User Label 0

Word 0-1: Record name of the highest readable record. For Masters, this is the highest record in the set (i.e. Capacity). For Details, this is the greatest number of records that have been written to the set thus far. For example, if there is room in the Detail data set for 100 records and 75 were written last week when the data set was loaded with DBLOAD, and yesterday 15 records were deleted from the data set, the "High Water Mark" is equal to a value of '75'.

Word 2-3: Number of unused records in the data set. This field is incremented when a record is deleted and decremented when a record is added. To determine the current number of entries used in the set subtract Word 1-2 (unused count) from Word 0-1 (capacity).

Word 4-5: The delete chain head for Details. This points to the record most recently deleted or contains a value of zero if no records have been deleted. This field is not used in Master data sets.

Data Set Records

The data in the data set records is arranged according to the Media records. These are formatted by the Schema Processor (DBSCHEMA).

CHAPTER 22 DISC FREE SPACE MAP

Disc Resident Data Structures

There are two disc resident free space data structures, the bit map and the descriptor table, for each disc volume that has a free space map, i.e. system discs and private volumes. The addresses of these data structures are kept in the disc label. The symbols that define the descriptor table and bit map are in the include file INCLDFS2.

Bit Map

The bit map is divided up into pages, which is the physical block of the map that is read or written. At the moment, a page is defined to be one sector (128 words) long, this may be changed by changing a compile time constant. The last word of the page is a checksum for that page, all other words are data. There is a one to one correspondence between bits in the map and sectors of the disc. A one bit represents a free sector and a zero bit represents an allocated sector. The bit map is a contiguous set of pages, enough to represent the entire disc, excluding spare tracks and spare sectors.

Descriptor Table (DT)

The descriptor table is an array of three word entries, one entry for each page of the bit map. Each entry looks like this:

```
                  ===================
                  =                                       =
word 0    =  largest space   =
                  =                                       =
                  ===================
                  =                                       =
word 1    =  starting space  =
                  =                                       =
                  ===================
                  =                                       =
word 2    =  ending space    =
                  =                                       =
                  ===================
```

Thus the descriptor table looks like this.

```
       ------------
=          = entry for page 0
       ------------
```

```
=          = entry for page 1
       ------------
=          = entry for page 2
       ------------
=          = entry for page 3
       ------------
            .
            .
            .
       ------------
=          = entry for last page
       ------------
```

Each entry describes the free space on the corresponding page of the bit map. The largest space word is the size of the largest contiguous block of free space on the page, which is not at the very beginning or very end of the page. That is, the first bit physically representing the space is not the first bit of data on the page or the last bit representing the space is not the last bit of data on the page. Starting space is the number sectors of contiguous space represented by the set of bits whose first bit is the first bit of data on the page. Ending space is the number of sectors of contiguous space represented by the set of bits whose last bit is the last bit of data on the page. The starting space and ending space fields allow looking across page boundaries, thus preventing fragmentation on page boundaries. Thus, if all sectors represented on a page are free, then starting and ending space will be the same and have the total number of free sectors represented on the page. Largest space will be zero, as there is no block of space that is not at the beginning or end of the page. A value of - 1 for all the fields in an entry indicates the corresponding page is bad, either from a checksum or I/O error.

Virtual Memory Resident Data Structures

For each system disc or physically mounted private volume there is a data segment which has information about the disc free space map, the current copy of the descriptor table, some work space for the procedures while in split stack mode and buffers for pages of the bitmap. The DST number of the data segment for a given disc is found in the LDTX entry for that disc.

Disc Free Space Data Segment

For each system disc or physically mounted private volume in the up and running system there is a DST which contains information about the disc free space map for that disc, some work area, a copy of the descriptor table and buffers for the pages of the bit map. All symbols that define these data segments are in the include file INCLDFS1, and they are prefixed with "ds'". The structure of the data segment is as follows:

```
          =================================
0 (%0) =             ds'ldev                =
```

```
          =---------------------------------=
 1 (%1) =             ds'dst                =
          =---------------------------------=
 2 (%2) =                                   =
          =--------- ds'disc'size ----------=
 3 (%3) =                                   =
          =---------------------------------=
 4 (%4) =       ds'last'page'of'map         =
          =---------------------------------=
 5 (%5) =      ds'last'buffer'index         =
          =---------------------------------=
 6 (%6) =                                   =
          =--------- ds'map'address --------=
 7 (%7) =                                   =
          =---------------------------------=
 8 (%10) =            ds'lock               =
          =---------------------------------=
 9 (%11) =         ds'lock'count            =
          =---------------------------------=
10 (%12) =         ds'queue'head            =
          =---------------------------------=
11 (%13) =         ds'queue'tail            =
          =---------------------------------=
12 (%14) =      ds'descriptor'table         =
          =---------------------------------=
13 (%15) =     ds'buffer'page'number        =
          =---------------------------------=
14 (%16) =        ds'buffer'dirty           =
          =---------------------------------=
15 (%17) =        ds'buffer'area            =
          =---------------------------------=
16 (%18) =    ds'first'threshold'page       =
          =---------------------------------=
17 (%21) =                                  =
          =-- ds'size'of'last'allocation  --=
18 (%22) =                                  =
          =---------------------------------=
```

```
          =---------------------------------=
19 (%23) =  ds'last'page'allocated'from    =
          =---------------------------------=
20 (%24) =     ds'next'buffer'index         =
          =---------------------------------=
21 (%25) =        ds'page'number            =
          =---------------------------------=
22 (%26) =        ds'word'number            =
          =---------------------------------=
23 (%27) =         ds'bit'number            =
          =---------------------------------=
24 (%30) =        ds'page'pointer           =
          =---------------------------------=
25 (%31) =    ds'starting'word'number       =
          =---------------------------------=
26 (%32) =    ds'starting'bit'number        =
          =---------------------------------=
27 (%33) =                                  =
          =----- ds'number'of'sectors  -----=
28 (%34) =                                  =
          =---------------------------------=
29 (%35) =         ds'bit'count             =
          =---------------------------------=
30 (%36) =        ds'entry'type             =
          =---------------------------------=
31 (%37) =       ds'buffer'index            =
          =---------------------------------=
32 (%40) =                                  =
          =--------- ds'disc'address -------=
33 (%41) =                                  =
          =---------------------------------=
34 (%42) =       ds'error'status            =
          =---------------------------------=
```

The rest of the data segment contains tables whose size and location is dependent on the size of the disc and or the number of buffers in the data segment. They are shown below just to demonstrate there relation to one another, for there actual location, the pointers should be examined. The symbol "ds'array'area" defines the start of the area. The first table is the descriptor table, it is in the same format as the disc copy, but a dummy entry of all zeros is added before and after the table, these are needed by procedures "Find'Page" and "Build'Descriptor'Entry". The pointer to this table is "ds'descriptor'table", it points to the entry for page zero, not the dummy entry.

```
==================================
=              0                 =
=--------------------------------=  dummy
=              0                 =
=--------------------------------=  entry
=              0                 =
==================================
=        largest space           =
=--------------------------------=  entry for
=        starting space          =
=--------------------------------=   page 0
=         ending space           =
==================================
=        largest space           =
=--------------------------------=  entry for
=        starting space          =
=--------------------------------=   page 1
=         ending space           =
==================================
                :
                :
                :
==================================
=        largest space           =
=--------------------------------=  entry for
=        starting space          =
=--------------------------------=  last page
=         ending space           =
==================================
=              0                 =
=--------------------------------=  dummy
=              0                 =
=--------------------------------=  entry
=              0                 =
==================================
```

The next table is ds'buffer'page'number table, it has a one word entry for each buffer in the data segment. Each entry contains the page number of the page currently in the corresponding buffer or -1 if the buffer is empty. This is pointed to by "ds'buffer'page'number".

```
==================================
=        buffer 0 entry          =
==================================
=        buffer 1 entry          =
==================================
                :
                :
```

```
                :
==================================
=       last buffer entry        =
==================================
```

The next table is the ds'buffer'dirty table, which has a one word entry for each buffer. A TRUE indicates the page in the corresponding buffer is dirty, i.e. the disc copy is not up-to-date. A FALSE indicates that the buffer is clean. If DFS was compiled with dirty buffer management turned off, this table is not present and the ds'buffer'dirty pointer is zero.

```
==================================
=        buffer 0 entry          =
==================================
=        buffer 1 entry          =
==================================
                :
                :
                :
==================================
=       last buffer entry        =
==================================
```

The remainder of the data segment contains the buffers, each buffer is the size of one page of the bit map, which is currently one sector(128 words). The beginning of the buffer area is pointed to by "ds'buffer'area" and the number of buffers is the value in "ds'last'buffer'index" plus one.

```
==================================
=                                =
=                                =
=                                =
=            buffer 0            =
=                                =
=                                =
=                                =
==================================
=                                =
=                                =
=                                =
=            buffer 1            =
=                                =
=                                =
=                                =
==================================
                :
```

```
                :
                :
==================================
=                                =
=                                =
=                                =
=           last buffer          =
=                                =
=                                =
=                                =
==================================
```

Each of the fields of the data segment is described in the include file INCLDFS1, where they are defined. It should be noted that the following fields are just workspace, used to pass information between procedures while in split stack mode and have no meaning between calls to the disc free space management subsystem:

```
     ds'page'number              ds'word'number
     ds'bit'number               ds'page'ptr
     ds'starting'word'number     ds'starting'bit'number
     ds'number'of'sectors        ds'entry'type
     ds'bit'count                ds'buffer'index
     ds'disc'address
```

The field ds'error'status normally has no meaning between calls unless the error'type field has a value greater than "fatal'dfs'error", in which case it means that disc space may no longer be allocated on this disc.

CHAPTER 23 MPE DISC CACHING

Disc Caching Overview

Disc Caching is an optional feature of MPE that utilizes excess main memory and excess CPU horsepower to keep portions of frequently referenced disc "domains" in memory. (A disc "domain" is a copy of a portion of disc residing in main memory. These disc domains are considered "cached" when they are in memory and are considered "mapped" when there is I/O pending against them.) Disc Caching manages the bi-directional transfer of these disc domains between main memory and disc storage. No main memory is permanently dedicated to cached disc domains. Cached disc domains share main memory with all other types of MPE segments and are not treated differently by the memory manager. By keeping cached disc domains in memory, a significant portion of the references to disc storage can be resolved without actually having to physically access the disc. Disc Caching policies are integrated into the MPE Kernel, File System, and I/O System which allows the system performance to be tuned based on the current workload and resource availability.

Disc Caching uses the MPE kernel resource management mechanisms and strategies. These mechanisms are extended to handle cached disc domains in the same manner as segments. Thus, cached disc domains can be of variable size, fetched in parallel with other segments or cached domains, garbage collected, and replaced in the same manner as stacks, data and code segments. The relative use of main memory between stacks, data and code segments, and cached disc domains is dynamic. This partitioning is based on the workload's current requirements and current memory availability.

Disc Caching can be enabled/disabled on a disc by disc basis. When caching is enabled for the first disc, the code segment containing the Disc Caching code will be locked into memory. Also at this time the Cache Directory Table (CDT) will be built and locked into memory. When caching is disabled for the last disc, the code segment will be unlocked from memory and the CDT will be released. Thus if caching is not enabled no memory will be wasted.

The CDT is used to keep track of the following information:

1) The disc ldevs currently enabled for caching. There will be a Device Entry in the table for each cached disc.

2) A linked list of cached domains for each disc with caching enabled. The head and tail of this linked list will be contained in the Device Entry. (I.e. there is a separate linked list of cached domains for each cached disc ldev.)

3) The cached domains that currently have user I/O pending (i.e. FREADS/FWRITES) or have memory management I/O pending (i.e. fetching the disc domain into memory, or posting the disc domain back out to disc). There will be a Mapped Domain Entry in the table for each disc domain has that I/O pending and is thus "mapped".

4) A linked list of all user I/O pending against the mapped disc domains. There will be a Logical Disc Request (LDR) queued to the Mapped Domain entries that will describe the user I/O to take place. This is analogous to a Disc Request queued to a specific DIT waiting for service.

When a request is made to access disc information, Disc Caching must first determine if the requested disc domain is present in memory. Disc Caching will first determine if the requested area of disc is already mapped into memory by scanning through the Mapped Domain entries of the CDT. If the requested transfer can be satisfied with a currently mapped disc domain, then the I/O request will be queued (FIFO) behind the other I/Os pending against that mapped domain. If the requested area is not already mapped, then a search is made through the linked list of cached disc domains for the specified disc ldev. (The region header contains the disc address and size that a disc domain represents.) If the requested domain is found in this list (i.e. present in memory), then this region will be mapped. A domain is then considered mapped when there is an entry for it in the Mapped Domain portion of the CDT. Mapping the domain allows Disc Caching to manage the I/O pending and/or currently active for a particular disc domain. Once the disc domain is mapped and present, the data can be moved between the process' data area and the mapped disc domain. The process can then continue executing without interruption or a process switch. The user/subsystem process for which the move is done will be charged with the CPU overhead.

When a request is made to read data that is not currently cached in memory (i.e. a read "miss"), the fetch strategy uses the File System's knowledge of the type of access (sequential or random), the extent size of the file, along with the current memory load to select the optimal size of the disc domain to be fetched and mapped into memory. The fetch of the disc domain is then initiated on the user's stack without a process switch. After the fetch is initiated, it completes in an unblocked manner so that this process (if no-wait I/O) or another process can proceed in parallel with the cache fetch.

In general, when writing, a process will not wait for completion of the physical I/O. Instead, the process will be awakened as soon as the transfer has completed between the process's data area and the mapped disc domain (i.e. no-wait-for-post). The physical I/O will then be posted at background priority while the process continues. (Users can specify wait-for-post on a file by file basis in place of the default no-wait-for-post with the FSETMODE intrinsic. This can be done on a global basis via :CACHECONTROL.) If the access request is a write and there is a current write pending against the specified mapped disc domain, the process request is queued until the pending write is posted to disc. If the disc domain to be written is not currently cached in memory, a free piece of memory will be obtained to map the corresponding disc image and then the "write" takes place from the process' data area to the mapped disc domain. This prevents data from having to be read before being written. After that, a post to disc is initiated (on any write only the portion of a mapped disc domain that is modified will be posted to disc). After the move to the mapped disc domain is complete and the post to disc is initiated, the process performing the "write" is allowed to continue to run without having to wait for the post to complete. Writes that must be posted to disc in a certain order use the Global Serial Write Queue. These ordered writes include things like updating disc free space maps for a new file extent before updating the file extent map in the file label.

There are two disc request entries used for disc caching requests. The first entry is a Logical Disc Request (LDR) entry and is used to manage the data moves to/from the user's data area and the disc domain (i.e. the logical I/O). The second entry is a regular Disc Request (DRQ) entry and is used to perform the physical I/O necessary to map a disc domain (for a read "miss") or to perform the physical post (on write requests). The disc domain will remain mapped until both the logical and physical I/O completes. If a request is not completely described by one disc domain already in memory or a Mapped Domain CDT entry (i.e. the requested disc area falls into more than one disc domain) then the overlapping disc domain(s) will be flushed to disc and the new complete disc domain will be fetched (if read) and mapped - no partial mappings are allowed.

The DST number of the Cache Directory Table (CDT) is at %1273 and the bank and offset are kept in %1274-%1275. The Caching Sir (2) is used when starting and stopping caching (via :STARTCACHE/:STOPCACHE) and by the LOADER when loading a program file (this sir is only used when updating the STT at load time).

When caching is enabled for a disc, a bit in the flags word of the DIT is set. Also, the Global Serial Write queue can be found by examining the header entry of the Disc Request Table. See Chapter 13 for a more detailed explanation of both the DIT and the Disc Request Table header. See Chapter 2 for a description of the Memory Region Header for a disc domain (cached region).

Disc Caching Tables Overview

```
               Cache
             Directory
               Table
               (CDT)
        ------------------------
        |                      |
        |        Header        |
        | (Info for table mgt) |
        |                      |
        |======================|
        |                      |
   +----|   Device entry # 1   |  ----------+
   |    |                      |            |
   |    |----------------------|            |
   |    |                      |            |
   |    |   Device entry # 2   |            |
   |    |                      |            |
   |    |----------------------|            |
   |    |          .           |            |
   |    |          .           |            |
   |    |          .           |            |
   |    |----------------------|            |
   |    |                      |            |
   |    |   Device entry # N   |            |
   |    |                      |            |
   |    |======================|            |
   |    |                      |<-----------+
 ..|....|   Mapped Domain # A  |===============> to 1st Logical
 . | +->|     for Dev # 1      |--+                   Disc
 . | |  |----------------------|  |                   Request
 . | |  |                      |  |                   Entry
 . | +--|   Mapped Domain # B  |<-+                   (LDR)
 . | +->|     for Dev # 1      |--+
 . | |  |----------------------|  |
 . | |  |                      |  |
 . | +--|   Mapped Domain # C  |<-+
 . |    |     for Dev # 1      |
 . |    |----------------------|
 . |    |                      |
 . |    |          .           |
 . |    |          .           |
 . |    |          .           |
 . |    |                      |
 . |     ------------------------
```

Memory Regions

```
. |
. |
. |    -------------------------
. |    |                       |
. +--->|   1st Cached Region   |
.      | (Cached disc domain)  |
. +----|       (Dev #1)        |<---+
. |    |                       |    |
. |    -------------------------    |
. |                                 |
. |    -------------------------    |
. |    |                       |    |
. +--->|   2nd Cached Region   |----+
.      | (Cached disc domain)  |
. +----|       (Dev #1)        |<---+
. |    |                       |    |
. |    -------------------------    |
. |                                 |
. |                .                |
. |                .                |
. |                .                |
. |                                 |
. |    -------------------------    |
. |    |                       |    |
. +--->| Mapped Cache Region   |----+
......>|(Mapped disc domain #A)|
  +----|        (Dev #1)       |<---+
  |    |                       |    |
  |    -------------------------    |
  |                                 |
  |                .                |
  |                .                |
  |                .                |
  |                                 |
  |    -------------------------    |
  |    |                       |    |
  +--->|   Last Cache Region   |<---+
       | (Cached disc domain)  |
       |       (Dev #1)        |
       |                       |
       -------------------------
```

Cache Directory Table

The Cache Directory Table (CDT) is the bookkeeping structure for managing cached disc domains. This table is divided into 3 parts:

CDT Header Entry
This entry contains all information necessary to manage the entire table and also contains global caching related information.

CDT Device Entry
There will be one of these entries for every disc ldev that currently has caching enabled. These entries keep track of all cached disc domains in memory for this device. In addition, these entries contain statistics regarding the number of I/Os performed to the ldev.

CDT Mapped Domain Entry
These entries describe disc domains that are currently "mapped" into memory. This means that there is logical I/O (cache move) and/or physical I/O (fetch or post) pending. These entries keep track of the state of the cached disc domain (IMI, ROC, etc.) just as the DST Table keeps track of data segments.

The following low core cells contain the address of the CDT:

%1273   contains the DST Number of the CDT
%1274   contains the Bank Number of the CDT
%1275   contains the Offset within the bank of the CDT

Header Entry

0	# Entries	CDT'ENTRIES
1	Entry Size (%30)	CDT'SIZE
2	# Free Entries	CDT'FREE'COUNT
3	1st Free Entry (table offset)	CDT'FREE'HEAD
4	Last Free Entry (table offset)	CDT'FREE'TAIL
5	Max # Entries Used	CDT'MAX'USED
6	# Ldevs cached	CDT'NUM'LDEVS
7	1st Cache device entry (entry number)	CDT'DISC'HEAD
%10	# Words this DST	CDT'DST'WORDS
%11	TRUE if stopcache pending	CDT'STOP'PND
%12	# Sectors sequential fetch	CDT'SEQ'MINFTCH
%13	# Sectors random fetch	CDT'RND'MINFTCH
%14	TRUE if wait for physical post	CDT'FORCE'POST
%15	Head of impeded queue (PIN)	CDT'STOP'QUEUE
%16		
	. . .	
%27		

CDT'ENTRIES
The total number of CDT entries configured in this table (i.e. includes all three types of entries). The number of entries in the table will be:

```
    1 entry for the header
  + 1 entry for each disc ldev configured.
        (CDT Device entries)
  + 1 entry for each DRQ configured.
        (CDT Mapped Domain entries)
```

This scheme insures that this table can never overflow (since an entry in the DRQ table is always obtained before an entry in this table).

CDT'SIZE
Size of each entry in the table.

CDT'FREE'COUNT
Total number of entries currently unassigned.

CDT'FREE'HEAD
Table relative offset (i.e. Entry number * entry size) of the first available entry.

CDT'FREE'TAIL
Table relative offset of the last available entry.

CDT'MAX'USED
The maximum number of entries in use at one time.

CDT'NUM'LDEVS
The number of ldevs currently cached.

CDT'DISC'HEAD
The entry number of the first Device Entry.

CDT'DST'WORDS
The total number of words in this data segment.

CDT'STOP'PND
This value will be TRUE if there is a pending :STOPCACHE.

CDT'SEQ'MINFTCH
If there is a prefetch for a sequential read ("miss"), the size of the prefetch is delimited by the extent size of the file. Within this limitation, the prefetch is equal to the greater of two sizes:
1) Requested size.
2) The largest integer multiple of the request size that is smaller than the value found in this cell.

The default value is 96 sectors. (This value may be changed via :CACHECONTROL).

CDT'RND'MINFTCH
This is the same as CDT'SEQ'MINFTCH except that it's for random access. The default value is 16 sectors. (This value may be changed via :CACHECONTROL).

CDT'FORCE'POST
When this value is TRUE, all writes will "block" until the physical update on disc completes. The system default is FALSE. (Can be altered via :CACHECONTROL).

CDT'STOP'QUEUE
If CDT'STOP'PENDING is TRUE this will be the PIN number of the head pin of the processes impeded until the :STOPCACHE completes.

<u>Device Entry</u>

0	Next ldev entry (entry number)	CDT'DE'NEXT'LDEV
1	Prev ldev entry (entry number)	CDT'DE'PREV'LDEV
2	Ldev for this disc	CDT'DE'LDEV
3	# Pages in device's domain	CDT'DE'MAPD'PAGES
4	# Disc domains currently mapped	CDT'DE'MAPD'CNT
5	Head of mapped domain (entry number)	CDT'DE'MAPD'HEAD
6	Tail of mapped domain (entry number)	CDT'DE'MAPD'TAIL
7	# Disc domain regions for this device	CDT'DE'REGIONS
%10	Memory address of head cached disc domain	CDT'DE'REG'HD
%12	Memory address of tail cached disc domain	CDT'DE'REG'TL
%14	# Read hits	CDT'DE'RHIT
%16	# Write hits	CDT'DE'WHIT
%20	# Read misses	CDT'DE'RMISS
%22	# Write misses	CDT'DE'WMISS
%24	# Stops	CDT'DE'STOP
%26	Memory address of last referenced domain	CDT'DE'SCANPT

CDT'DE'NEXT'LDEV
The entry number of the next Device Entry.

CDT'DE'PREV'LDEV
The entry number of the previous Device Entry.

CDT'DE'LDEV
The Ldev number for this cached device.

CDT'DE'MAPD'PAGES
Total number of main memory pages allocated to disc domains for this cached device. This includes mapped and unmapped regions. (1 main memory page = 128 words).

CDT'DE'MAPD'CNT
The total number of Mapped Domain entries associated with this Device Entry.

CDT'DE'MAPD'HEAD
The entry number of the first Mapped Domain entry for this device.

CDT'DE'MAPD'TAIL
The entry number of the last Mapped Domain entry for this device.

CDT'DE'REGIONS
The total number of disc domain regions for this ldev (includes mapped and unmapped regions).

CDT'DE'REG'HD
Memory address to the head region of the disc domain linked list. Disc domain regions are linked in order based on the disc address they represent (i.e. small disc address at head, large disc address at tail). This address will not point to the region base (RB), but to the next domain (ND) field of the region header. (This is to facilitate the use of the LLSH instruction).

CDT'DE'REG'TL
Memory address of the tail region of the disc domain linked list. This address will be of the previous domain (PD) field of the region header.

CDT'DE'RHIT
Total number of times that a read was requested and the requested disc domain was present in memory - i.e. a read "hit". This means that the read completed without performing any I/O (to fetch the domain). Thus this is actually the number of read I/Os eliminated. This value will reset to zero on overflow.

CDT'DE'WHIT
Total number of times that a write was requested and the requested disc domain was present in memory - i.e. a write "hit". If there was no other write pending to the "hit" domain, then the process would continue as soon as the cache move completes - thus eliminating a block for I/O. Otherwise, the process would block waiting for the first write to complete. This value will reset to zero on overflow.

CDT'DE'RMISS
Total number of times that a read was requested and the requested disc domain was not in memory - i.e. a read "miss". This means that the requested disc domain had to be fetched into memory before the read could complete - thus potentially blocking the process. This value will reset to zero on overflow.

CDT'DE'WMISS
Total number of times that a write was requested and the requested disc domain was not in memory - i.e. a write "miss". This does not mean that the process would block until the disc domain is fetched as is the case for reads. Rather, a free memory region would be obtained to be the destination of the cache move. This disc domain would then be posted in the background (unless overridden via :CACHECONTROL or FSETMODE) allowing the process to continue without blocking. This value will reset to zero on overflow.

CDT'DE'STOP
Total number of times that a process had to block on a cache transfer. Will reset to zero on overflow.

CDT'DE'SCANPT
The memory address of the last region looked at on a search. This address will be of the next domain (ND) field of the region header. This value will be used along with CDT'DE'REG'HD to determine where to start the next search for a cached disc domain At times it will be more efficient to start with this address since the disc domain requested may be of a higher disc address than found in this region header, rather than always starting the search with CDT'DE'REG'HD.

Mapped Domain Entry

```
      ----------------------------------------------
  0  | Prev mapped domain entry (entry number)| CDT'MD'PREV
     |----------------------------------------|
  1  | Next mapped domain entry (entry number)| CDT'MD'NEXT
     |----------------------------------------|
  2  |           Start sector                 | CDT'MD'SECTOR
     |-                                      -|
     |              address                   |
     |----------------------------------------|
  4  |            Last sector                 | CDT'MD'END'SECTOR
     |-                                      -|
     |              address                   |
     |----------------------------------------|
  6  | A| I| I| M| L| F| R| V| N| S| /|  S     | CDT'MD'FLAGS
     | B| M| M| I| O| W| O| I| O| E| /|  T     |
     | S| I| O| S| C| I| C| R| P| Q| /|  A     |
     | E|  |  | S| K| P|  | G| O|  | /|  T     |
     | N|  |  |  | E|  |  | I| S|  | /|  E     |
     | T|  |  |  | D|  |  | N| T|  | /|        |
     |----------------------------------------|
  7  |          # Reads pending               | CDT'MD'READ'CNT
     |----------------------------------------|
 %10 |         # Writes pending               | CDT'MD'WRITE'CNT
     |----------------------------------------|
 %11 |            Lock waiting                | CDT'MD'LKD'CDT
     |----------------------------------------|
 %12 |       Head of impeded LDR              | CDT'MD'IMPED'HD
     |----------------------------------------|
 %13 |       Head of active LDR               | CDT'MD'LDR'HEAD
     |----------------------------------------|
 %14 |          Memory address                | CDT'MD'MEM'ADR
     |-                                      -|
     |             if present                 |
     |----------------------------------------|
 %16 |   DRQ for this mapped domain           | CDT'MD'DISCREQ
     |----------------------------------------|
 %17 |          # Flushing CDTs               | CDT'MD'LK'CNT
     |----------------------------------------|
 %20 |   Ldev for this mapped domain          | CDT'MD'LDEV
     |----------------------------------------|
 %21 |   Head impeded queue (PIN)             | CDT'MD'IMPEDED
     |----------------------------------------|
 %22 |   Device entry (entry number)          | CDT'MD'DE
     |----------------------------------------|
 %23 |                                        |
     |                 .                      |
     |                 .                      |
     |                 .                      |
 %27 |                                        |
      ----------------------------------------
```

CDT'MD'PREV
Entry number of the previous mapped domain entry for this device.

CDT'MD'NEXT
Entry number of the next mapped domain entry for this device.

CDT'MD'SECTOR
The starting disc sector address representing this mapped domain entry.

CDT'MD'END'SECTOR
The ending disc sector address representing this mapped domain entry.

CDT'MD'FLAGS
Flags describing the state of this mapped domain entry and the region associated with it:

(0:1) - Absent.
Region is not present in memory.

(1:1) - IMI.
Region is already In-Motion-In. (Set when the fetch for this cached region is initiated).

(2:1) - IMO.
Region is In-Motion-Out. (Set by STARTOBJWRITE when performing the background post of a cached region).

(3:1) - MISS.
This disc domain was not present and had to be prefetched.

(4:1) - LOCK. Not used.

(5:1) - FWIP.
Forced Write In Progress. Region was forced out of memory to make room for another object.

(6:1) - ROC.
Recover Overlay Candidate. Region may be forced out of memory to make room for another object. However, if this region is referenced again it can be recovered.

(7:1) - VIRGIN.
Clean region in the write state. Cleared as soon as a move completes. (I.e. if this bit is on, then a write can complete immediately. Otherwise the write will have to wait until the current write completes the physical post).

(8:1) - NOPOST.
Set when the CDT is being posted out as a result of a write request that did not want to wait for the physical post to complete. This will be cleared by the cache completor when the physical post completes. (This is used to insure that a cache move for any subsequent write request will not be serviced until the physical post completes.)

(9:1) - SEQ.
Set if doing sequential I/O. When the request for the last area of this disc domain is complete, this domain will be made a ROC.

(10:3) - Not used.

(13:3) - STATE
0 - AVAIL. CDT is an available entry.

1 - READ. Only read LDR(s) are attached.
2 - WRITE. Write LDR(s) and possibly read LDR(s) are attached.
3 - FLUSH. CDT is being flushed out.
4 - LOCK. Unused.

CDT'MD'READ'CNT
The number of LDRs attached that are for reads (move not complete).

CDT'MD'WRITE'CNT
The number of LDRs attached that are for writes. NOTE: This count will not be decremented until both the cache move and the physical write completes. However, as soon as the cache move completes, the LDR will be dequeued from the CDT.

CDT'MD'LKD'CDT
Not used.

CDT'MD'IMPED'HD
The first LDR that is impeded. (I.e. the CDT is in a write state already and another write is attached. The second write will be placed in this queue until the first write completes.)

CDT'MD'LDR'HEAD
The first LDR that is on the active list for this CDT.

CDT'MD'MEM'ADDR
The memory address (region base) for this mapped disc domain, if present.

CDT'MD'DISCREQ
The disc request table index associated with this mapped disc domain. This will be used to fetch this region in, or to post this region after any logical I/Os (writes) have completed. (I.e. this DRQ is used for the physical I/O.)

CDT'MD'LK'CNT
Not used.

CDT'MD'LDEV
The ldev number for this mapped domain.

CDT'MD'IMPEDED
The PIN for the first process impeded on this mapped disc domain. Processes get impeded here when they do WAITFORIO when their LDR is on the CDT impeded queue and the Mapped Domain is currently being written out. (This will also happen upon a :STOPCACHE to force all LDRs to complete.) As soon as the physical post of the Mapped Domain is complete, all processes impeded here will be awakened.

CDT'MD'DE
The entry number for the Device entry that this Mapped Domain entry is associated with.

Logical Disc Request Table

%1017 Pointer to Logical Disc Request Table

NOTE:
This table is really part of the DRQ (Chapter 13). Any entry with the logical request bit set in the flags will conform to this format and not the format of the standard DRQ.

Logical disc requests entries are used to manage requests between the requesting process and a mapped disc domain. They are the counterpart of disc requests entries used to manage physical I/O requests between a process and a disc. These entries are kept as part of the DRQ Table, but will never be queued to the disc's DIT, instead they will be queued to the mapped disc domain CDT entry. LDR entries may only be placed onto the following queues:

1) The CDT active list.
2) The CDT impeded LDR list.
3) The Disabled Disc Request. (This will only happen if the buffer segment is absent when the logical I/O (cache move) is attempted.)

NOTE:
LDRs are singly linked onto the CDT queues and doubly linked onto the disabled disc request queue.

Logical Disc Request Entry

```
                           1 1  1  1  1  1
      3  4  5  6  7  8  9  0 1  2  3  4  5
   ------------------------------------------
 0 |//| S| I| B| D| D| S| C| M|/| C| D| L| I| LDR'FLAGS
   |//| B| O| L| O| O| E| D| O|/| U| I| D| N|
   |//| U| W| O| N|  | R| T| V|/| R| S| R|  |
   |//| F| A| C| E| P| I|  |  |/|  | A|  | L|
   |//|  | K| K|  | O| A| Q| D|/| R| B| R| O|
   |//|  | E| E|  | S| L| U| O|/| E| L| E| C|
   |//|  |  | D|  | T|  | E| N|/| Q| E| Q|  |
   |//|  |  |  |  |  |  |  | E|/|  | D|  |  |
   |-----------------------------------------|
 1 |        HODA of extent limit             | LDR'L'HODA
   |-----------------------------------------|
 2 |                Ldev                     | LDR'LDEV
   |-----------------------------------------|
 3 |   Mapped Domain CDT entry number        | LDR'CDT
   |-----------------------------------------|
 4 | S|       DST number                     | LDR'BUFDST
   |-----------------------------------------|
 5 |          Offset into DST                | LDR'BUFADR
   |-----------------------------------------|
 6 |  Strategy       |      Function         | LDR'STRAT'FUNC
   |-----------------------------------------|
 7 |    Count/Xlog/Control returns           | LDR'COUNT
   |-----------------------------------------|
%10 |                P1                      | LDR'PARM1
   |-----------------------------------------|
%11 |                P2                      | LDR'PARM2
   |-----------------------------------------|
%12 |                 | Qualifier | Status    | LDR'STATQ
   |-----------------------------------------|
%13 |              PIN number                | LDR'PCB
   |-----------------------------------------|
%14 |  Prev. LDR in queue (table relative)   | LDR'PREVQ
   |-----------------------------------------|
%15 |  Next  LDR in queue (table relative)   | LDR'NEXTQ
   |-----------------------------------------|
%16 |       HODA of extent base              | LDR'B'HODA
   |-----------------------------------------|
%17 |       LODA of extent base              | LDR'B'LODA
   |-----------------------------------------|
%20 |       LODA of extent limit             | LDR'L'LODA
   -------------------------------------------
```

LDR'FLAGS
Flags.

(0:3) - Not used.
(3:1) - <u>SBUF</u>.
Set if request is to/from a System Buffer.
(4:1) - <u>IOWAKE</u>.
Set if system should wake up the process when the logical I/O completes.
(5:1) - <u>BLOCKED</u>.
Set if the process wants to wait for the logical disc request to complete.
(6:1) - <u>DONE</u>.
Set when the logical disc request is complete and the process will be awakened (if IOWAKE is set)
(7:1) - <u>DO'POST</u>.
Set if the caller wants to be waited until the physical post to disc completes. Only valid for write requests.
(8:1) - <u>SERIAL'POST</u>.
Set when the physical post should be through the Global Serial Write queue.
(9:1) - <u>CDT'QUEUED</u>.
This request has been queued - either onto the CDT active queue (see CDT Mapped Domain entries) or onto the disabled disc request list.
(10:1) - <u>MOVE'DONE</u>.
The move has been completed, but the process won't be awakened until the DONE bit is set.
(11:1) - Not used.
(12:1) - <u>CUR'REQ</u>.
Set if this request is the current/active request.
(13:1) - <u>DISABLE</u>.
Set if the request is disabled.
(14:1) - <u>LDR'REQ</u>.
Set if this is a logical disc request.
(15:1) - <u>LDR'INLOC</u>.
Set if Mapped Domain CDT entry is in process's locality list.

LDR'L'HODA
The High Order Disc Address of the extent limit. (See note with LDR'B'HODA).

LDR'LDEV
The ldev for this request.

LDR'CDT
The CDT number for the Mapped Domain entry associated with this request.

LDR'BUFDST
Data Segment number for the target of the logical I/O request. If bit zero is set, then this is the process's stack.

LDR'BUFADR
Offset within the DST (above) for the target address. If the DST is the process's stack, then this address will be DB relative.

LDR'STRAT'FUNC
(0:8) - Strategy

0	- Unknown caller
1	- Unknown File System
2	- Spooler
3	- Directory
4-7-	Unknown caller
8	- Genmessage
9	- File System, Quiesce I/O
10	- File System, Sequential, No Buf
11	- File System, Direct, No Buf
12	- File System, Sequential, Buffered
13	- File System, Direct, Buffered
14	- File System, KSAM
15	- File System, IMAGE

(8:8) - Function

0	- Read
1	- Write

LDR'COUNT
On initiation, this specifies the requested transfer count (+words, -bytes). At completion of the request, this contains the actual transmission count (+words, -bytes).

LDR'PARM1
This is the High Order Disc Address of the requested disc sector.

LDR'PARM2
This is the Low Order Disc Address of the requested disc sector.

LDR'STATQ
Uniform status returns.

LDR'PCB
PIN of the requesting process.

LDR'PREVQ
Table relative index of the previous LDR in the queue. (NOTE: LDRs are singly linked on the CDT queues, and doubly linked on the disabled disc request queue).

LDR'NEXTQ
Table relative index of the next LDR in the queue.

LDR'B'HODA
The High Order Disc Address of the extent base. (Used when the logical disc request is through the file system. Caching uses this information when searching memory for a "hit" on a cached domain).

LDR'B'LODA
The Low Order Disc Address of the extent base. (See note above).

LDR'L'LODA
The Low Order Disc Address of the extent limit. (See note above).

READER COMMENT SHEET

MPE V Tables Manual for MPE V/E, Version G.00.00

32033-90010 September 1984

We welcome your evaluation of this manual. It is one of several that serve as a reference source for HP 3000 Computer Systems. Your comments and suggestions help us to improve our publications and will be reviewed by appropriate technical personnel. HP may make any use of the submitted suggestions and comments without obligation.

Is this manual technically accurate?	**Yes [] No []**	(If no, explain under Comments, below.)
Are the concepts and wording easy to understand?	**Yes [] No []**	(If no, explain under Comments, below.)
Is the format of this manual convenient in size, arrangement and readability?	**Yes [] No []**	(If no, explain or suggest improvements under Comments, below.)

Comments:

We appreciate your comments and suggestions. This form requires no postage stamp if mailed in the U.S. For locations outside the U.S., your local HP representative will ensure that your comments are forwarded.

Date:

FROM:

Name

Company

Address

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1070 CUPERTINO, CALIFORNIA

POSTAGE WILL BE PAID BY ADDRESSEE

Documentation Manager/47U-91
Hewlett-Packard Company
Computer Systems Division
19447 Pruneridge Avenue
Cupertino, California 95014

FOLD FOLD

Part No. 32033-90010
Printed in U.S.A. 09/84
E2412

HEWLETT
PACKARD